

# Pandemic Project Report



**Prepared by  
Group 16: Pandem Inc.  
Alexander Pieczynski, Ledio Sinjari and Trevor Evans**

**University of Illinois Chicago**

**September 2016**

# **Table of Contents**

[How to Use This Document](#)

[Table of Contents](#)

[List of Figures](#)

[List of Tables](#)

[Project Description](#)

[Project Overview](#)

[The Purpose of the Project](#)

[The User Business or Background of the Project Effort](#)

[Goals of the Project](#)

[Measurement](#)

[The Scope of the Work](#)

[The Current Situation](#)

[The Context of the Work](#)

[Work Partitioning](#)

[Competing Products](#)

[The Scope of the Product](#)

[Scenario Diagram\(s\)](#)

[Product Scenario List](#)

[Individual Product Scenarios](#)

[Stakeholders](#)

[The Client](#)

[The Customer](#)

[Hands-On Users of the Product](#)

[Priorities Assigned to Users](#)

[User Participation](#)

[Maintenance Users and Service Technicians](#)

[Other Stakeholders](#)

[Mandated Constraints](#)

[Solution Constraints](#)

[Implementation Environment of the Current System](#)

[Partner or Collaborative Applications](#)

[Off-the-Shelf Software](#)

[Anticipated Workplace Environment](#)

[Schedule Constraints](#)

[Budget Constraints](#)

[Naming Conventions and Definitions](#)

[Definitions of Key Terms](#)

[UML and Other Notation Used in This Document](#)

[Data Dictionary for Any Included Models](#)

[Relevant Facts and Assumptions](#)

[Facts](#)

[Assumptions](#)

[Requirements](#)

[Product Use Cases](#)

[Use Case Diagrams](#)

[Product Use Case List](#)

[Individual Product Use Cases](#)

[Functional Requirements](#)

[Data Requirements](#)

[Performance Requirements](#)

[Speed and Latency Requirements](#)

[Precision or Accuracy Requirements](#)

[Capacity Requirements](#)

[Dependability Requirements](#)

[Reliability Requirements](#)

[Availability Requirements](#)

[Robustness or Fault-Tolerance Requirements](#)

[Safety-Critical Requirements](#)

[Maintainability and Supportability Requirements](#)

[Maintenance Requirements](#)

[Supportability Requirements](#)

[Adaptability Requirements](#)

[Scalability or Extensibility Requirements](#)

[Longevity Requirements](#)

[Security Requirements](#)

[Access Requirements](#)

[Integrity Requirements](#)

[Privacy Requirements](#)

[Audit Requirements](#)

[Immunity Requirements](#)

[Usability and Humanity Requirements](#)

[Ease of Use Requirements](#)

[Personalization and Internationalization Requirements](#)

[Learning Requirements](#)

[Understandability and Politeness Requirements](#)

[Accessibility Requirements](#)

[User Documentation Requirements](#)

[Training Requirements](#)

[Look and Feel Requirements](#)

[Appearance Requirements](#)

[Style Requirements](#)

[Operational and Environmental Requirements](#)

[Expected Physical Environment](#)

[Requirements for Interfacing with Adjacent Systems](#)

[Productization Requirements](#)

[Release Requirements](#)

[Cultural and Political Requirements](#)

[Cultural Requirements](#)

[Political Requirements](#)

[Legal Requirements](#)

[Compliance Requirements](#)

[Standards Requirements](#)

## [Design](#)

[System Design](#)

[Design goals](#)

[Current Software Architecture](#)

[Proposed Software Architecture](#)

[Overview](#)

[Class Diagrams](#)

[Dynamic Model](#)

[Subsystem Decomposition](#)

[Hardware / software mapping](#)

[Data Dictionary](#)

[Persistent Data management](#)

[Access control and security](#)

[Global software control](#)

[Boundary conditions](#)

[Subsystem services](#)

[User Interface](#)

[Object Design](#)

[Object Design trade-offs](#)

[Interface Documentation guidelines](#)

[Packages](#)

## Class Interfaces

### Test Plans

Features to be tested / not to be tested

Pass/Fail Criteria

Approach

Suspension and resumption

Testing materials ( hardware / software requirements )

Test cases

Testing schedule

### Project Issues

Open Issues

Off-the-Shelf Solutions

Ready-Made Products

Reusable Components

Products That Can Be Copied

New Problems

Effects on the Current Environment

Effects on the Installed Systems

Potential User Problems

Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Follow-Up Problems

Tasks

[Project Planning](#)

[Planning of the Development Phases](#)

[Migration to the New Product](#)

[Requirements for Migration to the New Product](#)

[Data That Has to Be Modified or Translated for the New System](#)

[Risks](#)

[Costs](#)

[Waiting Room](#)

[Ideas for Solutions](#)

[Project Retrospective](#)

[Glossary](#)

[References / Bibliography](#)

[Index](#)



## List of Figures

[Figure 1 - Pandemic Context](#)

[Figure 2 - Implementation Environment Diagram](#)

[Figure 3 - Faction Dependent Use Case Diagram](#)

[Figure 4 - Faction Independent Use Case Diagram](#)

## List of Tables

### [Table 1 - Pandemic Business Events List](#)

## **Project Description**

### **Project Overview**

The Game is spread between two player factions - BioHydra and VacSect - which are fighting a global war on bioterrorism. The player may join either faction, electing to either spread viruses or wipe them out. The game is played by moving around in the real world to locate or spread viruses, then playing a minigame to interact with them. Each faction has a slightly different play experience, but ultimately plays the same game.

### **The Purpose of the Project**

#### **The User Business or Background of the Project Effort**

The growth of augmented reality in the mobile space has allowed for new innovations in the world of gaming. To adhere to the specific strengths of our hardware, the mobile phone, we aim to create an augmented reality location based game on a global scale. **Pandemic**, much like the epidemiology it is inspired by, will spread across continents and live on the phones of players all over the world.

There are many TV shows, games, and movies that target bio terrorism, but none of them really involve the user in the action. Additionally, there is a stark few games that target the more benevolent side of a virus outbreak: the people stopping it in its tracks. We seek to create a game that allows the player to really interact with the world around them as they spread or defeat their viruses, and see how their plague spreads.

A widespread problem among games is that they don't involve the player personally in the experience. While this makes sense for role playing games, where the player is taking on the role of another person, or for abstract games, where there are no human characters, it takes away from the experience in games where the player him- or herself is the protagonist. In **Pandemic**, the player will become the character in this global war.

### **Goals of the Project**

The goal of this product is to provide users with a fun and engaging experience that they can enjoy for a long time. Players will compete with each other for world domination, and feel strength in their unity.

We want the game to be playable anywhere in the world (within the limits of internet connection), so that players can grow their empires with friends across oceans, and the product can reach as many people as possible. Of course, we also want the game to be fun.

## **Measurement**

Initially, the number of downloads the game has would be an ideal measurement of success. To measure how fun the game is, we will consider both the user growth rate as well as user retention. We will also measure gameplay mechanic effectiveness (how geographically far from its' origin a virus has moved) to measure success.

A global product will have, of course, a global market. The wider the range of playability, the greater the number of potential customers, so we will consider the platform(s) the game is available on as well as the spread of customers across the globe.

## **The Scope of the Work**

### **The Current Situation**

Currently, there exists no real world location-based game involving spread of influence. Pandemic will close that gap and define a new genre of games that can come to the market.

Part of the fun of games is in the novelty. Players want to experience something new from time to time, and one of the most impactful ways to do that is to create a whole new genre. We hope that this will be a successful venture, so that we and other developers can expand on the topic and create more games of its kind.

### **The Context of the Work**

Pandemic requires that the location of the user be known to the program, as well as a map of the world. Without these things, virus and vaccine tracking will be impossible. Additionally, the game will require a server to store and load account and virus/vaccine information.

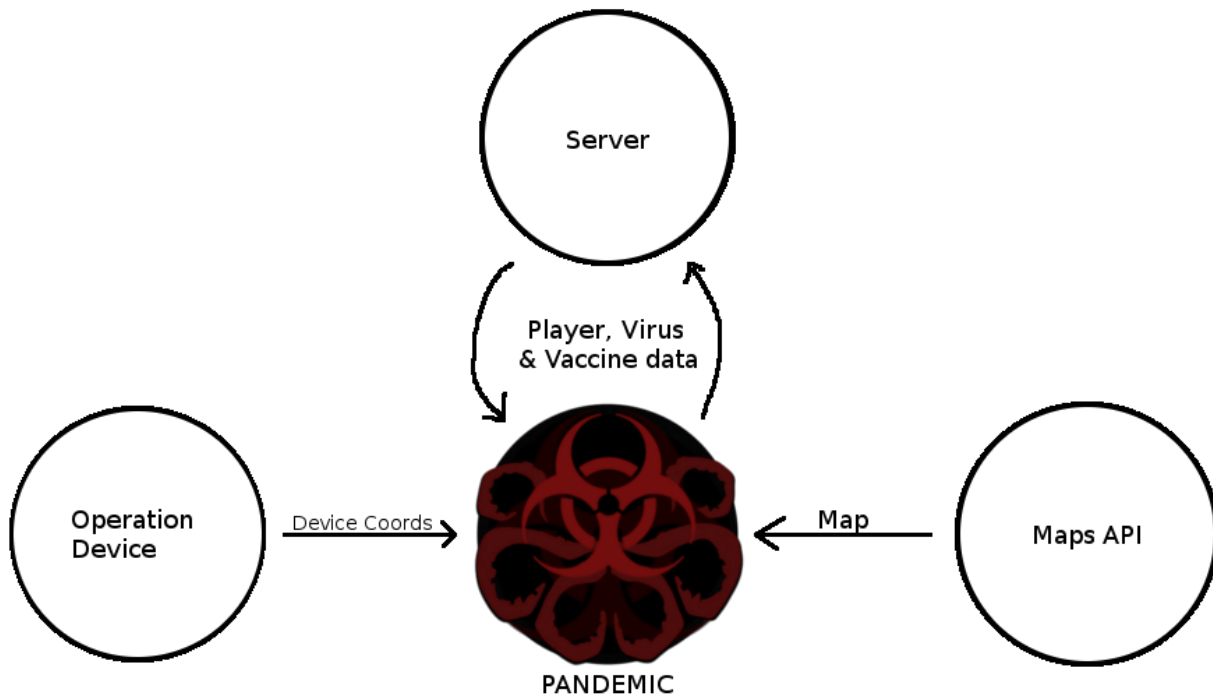


Figure 1 - Pandemic Context

**Work Partitioning**

**Business Event List**

**Event Name**

1. A user joins a faction

**Input and Output**

New user creates an account and petitions to join a team(in)

**Summary**

Team selection will at first be voluntary until there is an unbalance in teams. Then users will be shuffled to join the opposing faction.

2. A user is infected	Location data and virus data (in)	The user is infected with a strain and must choose to reverse engineer it to obtain antibodies, or facilitate its growth and spread it.
3. A user drops a strain	Viral / antiviral strain (out) location data (in)	Based on the location data, a user chooses an ideal location to spread the strain. A hotspot is created that radiates out according to environmental data.
4. A user of the opposing faction encounters a hotspot	Viral Data (in) Location data (in)	A mini game is initiated where the user must solve the challenge in order to remove the hotspot from play
5. A user levels up	Experience data(in)	A veteran user with a long running strain gains experience as the strain evolves and mutates, eventually leading to level ups and perks.
6. A user wins the mini-game	Points awarded to user(out)	Once a user is able to solve the challenge, they should be awarded experience points which get added to a total tally and perks which they can reinvest into their strains.
7. A high level user creates a strain	User's design decisions(in)	A higher level user is able to access the create feature to create a strain from scratch rather than customize an existing one.

8. A user changes factions.	Experience data / faction information (out)	A user will have the ability to change factions, after a waiting period their status will have changed to reflect an equal but opposite rank in the opposing faction.
9. A user tracks the progress of their strain	GPS data(in) Location history(in)	A user will be able to click on each strain they are involved with and view the progress through analytics tools such as heatmaps.
10 A strain can be modified with upgrades.	Strain information(in)	A user will be able to open a strain's profile and tweak it to change incubation time, transmission. The strain will need to be redeployed as a new mutation.
11. Global analytics	Location data (in) Game statistics(in) GPS data(out)	A user will be able to open and view global analytics in-app including which team is winning, how many members are in each team, hot spots and points of interest.

*Table 1 - Pandemic Business Events List*

## Competing Products

There are certainly many location-based games on the market today, but none of them are quite like **Pandemic**. Other location-based games such as Pokemon Go have proven the genre to be very successful in engaging players and getting them to go outside and meet other players. **Pandemic** is unique in that it takes advantage of the newer location-based possibilities of mobile phone gaming while providing the player a totally new type of game.

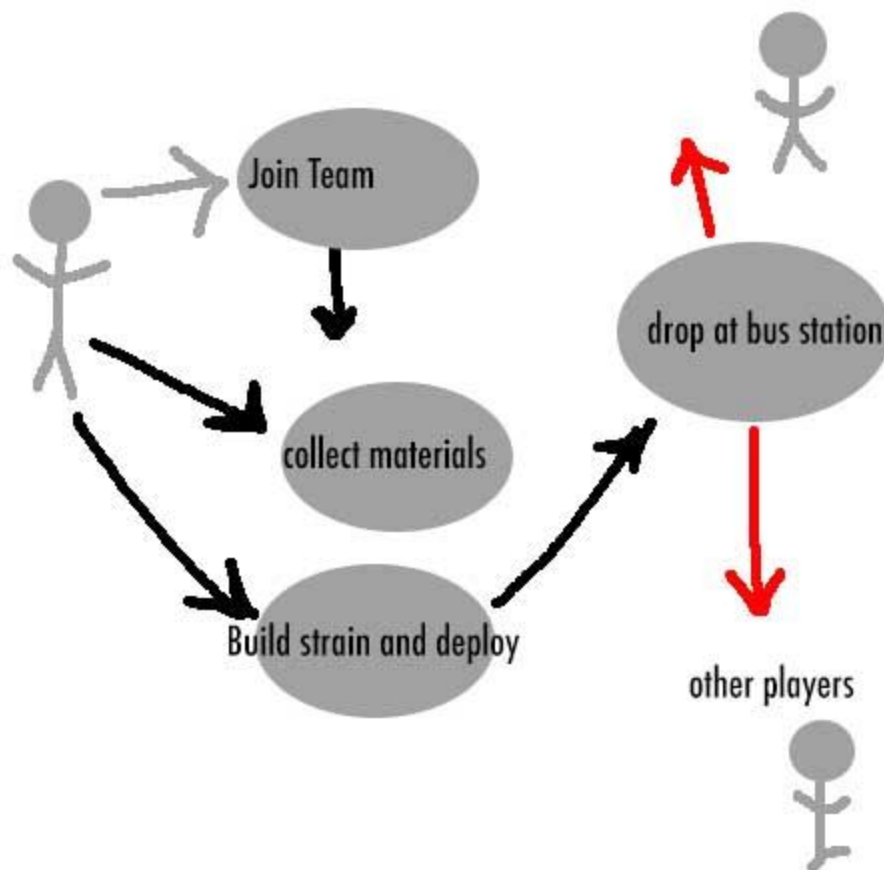
Another aspect of Pokemon Go that was successful was its use of teams. By having players join factions, the game fostered camaraderie between players of the

same team and made them more excited about the game. **Pandemic** takes it a step further by making the game completely team-based. Having pride in one's team causes users to want to play more often and recruit more friends to join their side.

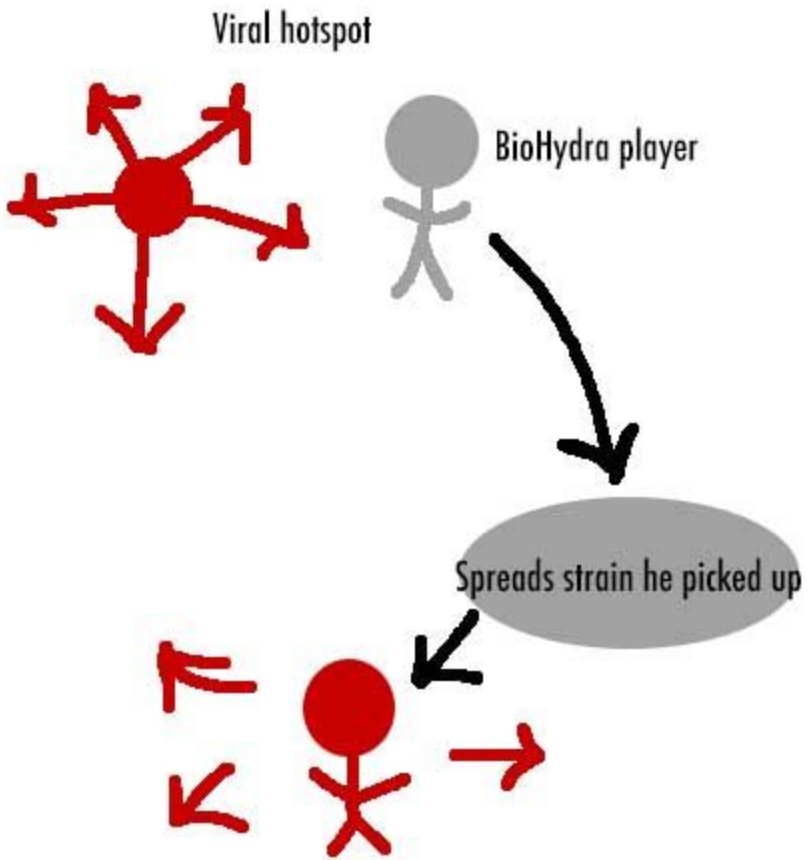
### The Scope of the Product

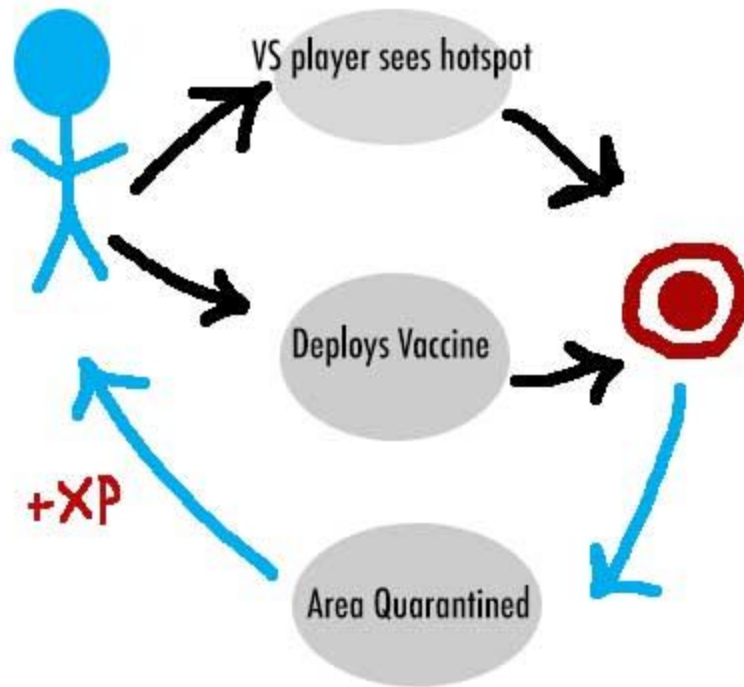
**Pandemic** is intended to be an augmented reality mobile game on a global scale. It will integrate with modern mobile hardware and take advantage of cellular antennae, GPS, and bluetooth protocols to interact with other devices. The application will also take advantage of existing software infrastructure, such as location databases provided by google maps to identify points of significance in the map. The application will be contained in servers with the front-end stored locally on cell phones. The software will accept direct user interaction into his or her phone.

### Scenario Diagram(s)









### Product Scenario List

A new player joins BioHydra, collects raw materials from a point of interest, and deploys a strain at his local bus station.

A low level player within the same BioHydra faction walks past the strain and purposefully infects himself, he then makes contact with a player of the opposing faction and infects him.

A high level player of VacSect designs and deploys an antibody strain into an existing infected area, effectively quarantining it and curing all those that come into contact.

A low level VacSect player is attacked by a low level BioHydra player and must fight off the infection.

A low level VacSect player has unknowingly walked into a viral hotspot and failed to make any quarantine measures, he has become infected and is unknowingly incubating a virus.

A high level BioHydra player has deployed a strain to a major bottling plant, that strain will now spread to that plant's major distribution points and infect people across the country.

A low level VacSect player has found a primordial strain but he is at too low a level to engineer it and must securely transport it to another member of the faction.

### **Individual Product Scenarios**

Mike downloads **Pandemic** and is prompted with the team recruitment screen. After viewing the propaganda campaigns, he decides to join BioHydra. His local minimap is updated with important locations and points of interest. Mike goes outside and starts visiting some locations, finding an in-game item (diseased possum carcass). Being at too low a level to engineer the roadkill into a proper viral weapon, Mike chooses to deploy it to his local Greyhound station. As a low level item with a small range, Mike is only able to infect one person before the hotspot is quickly vaccinated by a VacSect player. Mike gains 25 XP points from the event.

Scott is a level 2 BioHydra player that is doing his weekly rounds at the grocery store. He turns on the app and notices that a level 3 strain has been deployed near the fruit stand. Scott uses the cooperative feature to accept the strain (bypassing the infection minigame) and now becomes an active carrier. After finishing his grocery shopping and accumulating the adequate amount of steps, Scott's virus has fully incubated and he is able to infect others. He notices the cashier is a VacSect player and pings her device, triggering the infection minigame.

Jamie is a high level VacSect player and **Pandemic** veteran. She is traveling and has a layover at O'Hare. She notices that Terminal D is a viral hotspot and she engineers an antibody strain. After triggering the infection minigame, she wins, effectively quarantining the virus. She then deploys the vaccine to the location.

Jon is a novice VacSect player and has just started. He hasn't activated any features of the game so far and starts walking around to points of interest. He sees an area of the map with red waves radiating outward and walks towards it. His device receives a ping to fight off the infection, but he ignores it and walks in anyway. He picks up the strain at the hotspot but has become an unknowing carrier. He must now find another VacSect member, hand off the strain he picked up, and get cured.

Kurt is a high level BioHydra player living a rural area. He has accumulated various rare raw materials and engineers a strain with a long incubation period, extreme weather resistance, and a water-borne transmission. He decides to deploy his strain a bottling plant 15 miles away for a major soft drink distributor. After touring the facility and hitting deploy, the server maps that strain to (depending on the strain's variable success rate) all the locations that the bottling plant distributes to. After the incubation period, the server pings those areas again to make them active hotspots. With the initial drop point miles away, and a long incubation period, the virus runs rampant across players visiting supermarkets in that state.

Jennifer is a low level VacSect player that has stumbled onto the primordial strain while taking a tour of the same bottling plant the following day. She is at too low a level to be able to reverse engineer something with the same range, but knows from the analytics pages that this primordial relates to Kurt's high level virus. She must transport the strain to other VacSect members in her network.

## **Stakeholders**

### **The Client**

The primary client requesting the development of **Pandemic** is Dr. John T. Bell, a Computer Science professor at the University of Illinois at Chicago.

### **The Customer**

The target audience for **Pandemic** is people in the 18-30 year age range. These people will tend to have more time to devote to the game and so will be more satisfied with their results. **Pandemic**, much like Pokemon Go, will also tend to favor those who live in more populous areas as there are more people to spread your Virus/Vaccine to. However, the goal of **Pandemic** is to spread your influence outside of the places that you normally visit. Features will be put into the game to allow users in more rural areas to have just as much opportunity to spread their influence as those in cities.

## Hands-On Users of the Product

### Early Adopters / Reviewers

These users will be the first to download **Pandemic** and serve in an open beta to refine the product before it is ready for wide release.

Early adopters are typically very tech literate and look forward to exploring every feature and addition to the game. They accept bugs as part of the growing pains, and they build hype for the product into the general release.

Early adopters want to set trends and be ahead of new innovations, therefore it is essential for **Pandemic** to give them something new. **Pandemic's** team dynamics, location integration, and unique gameplay entice them to begin and continue playing.

Their responsibilities will be to drive hype for **Pandemic** and ensure that once the general release is ready, it debuts to a large existing user base and a large queue of eager new participants.

Early game adopters tend to be males between 18 - 30 that spend an inordinate amount of time on internet communities driving buzz about technology.

### Enthusiast / Hardcore gamers

Hardcore gamers are traditionally stationed to larger consoles, and attracting them to the mobile platform will be a challenge of itself, but they will be the lifeblood that drives **Pandemic**.

They will be responsible for driving progress the game ecosystem forward as well as growing the user base after the release. Hardcore gamers will be the players that reach the maximum levels early and set the pace of the game for the rest of the users. Hardcore gamers will also be the most dedicated in user retention, so they will be essential in networking their colleagues to join **Pandemic**.

Hardcore gamers are video game veterans that have been exposed to a wide variety of genres. They are extremely technically literate and look forward to bring energy and competition to the platform.

Hardcore gamers tend to be males between the early teens and early thirties.

## Casual Gamers

Casual gamers will build up the majority of **Pandemic's** user base. They are the ideal group in the growth sector because of sheer volume. Casual gamers will follow the hype set forth by the early adopters and join **Pandemic** in droves. However, they typically have a low retention rate so it will be essential to design an attractive and navigable user interface that will appeal to casuals.

Casual gamers will typically serve as the game's primary carriers, since early on they will be at too low a level to design their own strains. As the foot soldiers, they will mainly travel and spread existing strains, which will appeal to them because of the low learning curve, and emphasis on traveling and socializing.

Casual gamers tend to have limited technical literacy and low to medium technology experience. They will be Male and Female ages 12 and up of varying backgrounds.

## Priorities Assigned to Users

Key users: Casual gamers

Casual gamers make up the majority of users in a game community and **Pandemic** needs as large a user base as possible to function as intended. Thus appealing to casual gamers is the number one priority of the game. **Pandemic** should be engaging while providing a light learning experience. The rewards system should reward dedicated players but at the same time not isolate and punish players that don't log in daily.

Secondary users: Hardcore gamers

We would like to appeal to this demographic, but we feel that mobile games will never fully appeal to the hardcore gamer. **Pandemic** will be rich in features, but will likely not stand up to the amount of complexity hardcore gamers require while still appealing to a mass audience.

Unimportant users: Early adopters

What will primarily appeal to early adopters are new features and gameplay mechanics. We feel that those offered on release of **Pandemic** will be enough to generate enough buzz. Once the regular user base grows, however, it will be more important to maintain and refine current features than to always be on the cutting

edge. At this point we feel early adopters will have moved on to a new product and should hold little value to the core game community.

## **User Participation**

### Early Adopters : Two weeks before public release

We expect early adopters to sign up for a closed Beta and provide early feedback on playability, balance, and compatibility. As well as potential exploits and logic gaps that could be taken advantage of and must be patched. They will provide some usability requirements and thoughts on the UX. We expect these users to be very engaged in the first two weeks.

### Hardcore Gamers : Twenty plus hours a week

We expect a small dedicated sect of the **Pandemic** community to spend at least twenty hours per week and upwards of forty hours per week plaything the game. These users will be the first to enter the later stages of the game and will provide data on how balanced the systems are.

### Casual Gamers : Several hours a week

Casual Gamers will provide some of the basic use cases to analyze early on, as they will use the game the way most people intend to use the game. The information they provide, combined with their low level of participation and low technical literacy will provide crucial information on playability and fun that **Pandemic** will offer.

## **Maintenance Users and Service Technicians**

Game admins :

**Pandemic** gameplay will need to be supported by gameplay administrators.

They will be part of the development team and keep providing new features as well as balancing and rebalancing features to follow the flow of gameplay. Difficulty, experience, rewards, and other interactive features will consistently need to be tweaked in order to ensure optimal gameplay as the game spreads to a global user base.

System admins :

As with competing products such as **Pokemon Go** , being a large scale global game, **Pandemic** will need to be hosted on a cloud server solution. This will require system administrators to monitor network load and ensure accurate location data.

## Other Stakeholders

- Testers

**Pandemic** will need to go through rigorous testing before it can be released. This is due to the complexities of designing and deploying such an ambitious project. Although everything on **Pandemic** will run server side, it will need to be compatible across multiple mobile platforms and their respective development stacks. We will need to employ android and IOS testers across multiple iterations of both operating systems.

We will also need functional testers to go out and play the game to ensure proper location mechanics. We will then hold a closed public beta in order to reach and seed multiple parts of the world to ensure that the game will work correctly in specific locations as per the requirements.

- Business analysts

We will need B.A's to gather requirements from the third party software that will need to integrate into the **Pandemic** engine. Primarily this will be someone to interface with google maps location data and someone to interface with amazon AWS cloud services.

- System designers

We will need mobile UX designers to ensure that the game has an attractive, simple, yet powerful UI. Because we intend to appeal to casual gamers, UX designers will need to create a product with maximum usability.

- Marketing experts

Because a large user base is crucial to **Pandemic** we need marketing and social media experts to come up with a strategy to promote the game.

- Representatives of external associations



As mentioned previously, we hope to integrate **Pandemic** with Google location data as well as Amazon AWS services. We will need cooperation with both of those companies in order to have a successful endeavour.

## **Mandated Constraints**

### **Solution Constraints**

Description: The game will be mobile.

Rationale: The strength of a location based game lies in the ability to constantly change locations, thus the game must be played on a mobile device.

Fit criterion: All features of the game should be played in their ideal, designed and intended manner across mobile devices running both Android and IOS operating systems.

Description: The game shall be location based.

Rationale: In order for the primary gameplay mechanics of **Pandemic** to function, the game will need a stream of new location data always coming in.

Fit criterion: The product should take advantage of the cell towers and built in GPS, as well as Google location services to pinpoint both coordinates and related points of interest.

Description: The game will offer teams the ability to compete and level up.

Rationale: The game must engage users enough to keep playing by rewarding their efforts.

Fit criterion: All users will start at level 1 and gain experience through designing and spreading their strain, when they earn enough XP points, they will level up.

Description: The game must always be connected to the internet.

Rationale: **Pandemic** will need constant fresh data to function correctly, stale data will not be accepted by the game engine, thus we need the game always connected.

Fit criterion: whether through mobile data by a cell provider, or WiFi, the game should always have some connectivity and should always be pinging the location of the user. Players need to be constantly tracked, even while the app is off (background thread) in order to incubate, spread, and catch new strains.

Description: The servers must store persistent location data.

Rationale: The player will gain power boosts depending on locations he visits. A hospital, airport, factory, farm can all offer ways to spread a strain with their own unique strengths and weaknesses.

Fit criterion: All of these points of interest must be aggregated and stored server side. Once a user creates a hotspot, that location data must be stored server side and become visible to all other players. All other players need to be able to interact with user generated location data.

Description: Players need to be able to interact with each other.

Rationale: A good location game will facilitate interaction and cooperation between local players.

Fit criterion: The server must track and sort users that are near each other and make them visible. A player should be able to recognize a teammate or enemy near their hotspot.

## **Implementation Environment of the Current System**

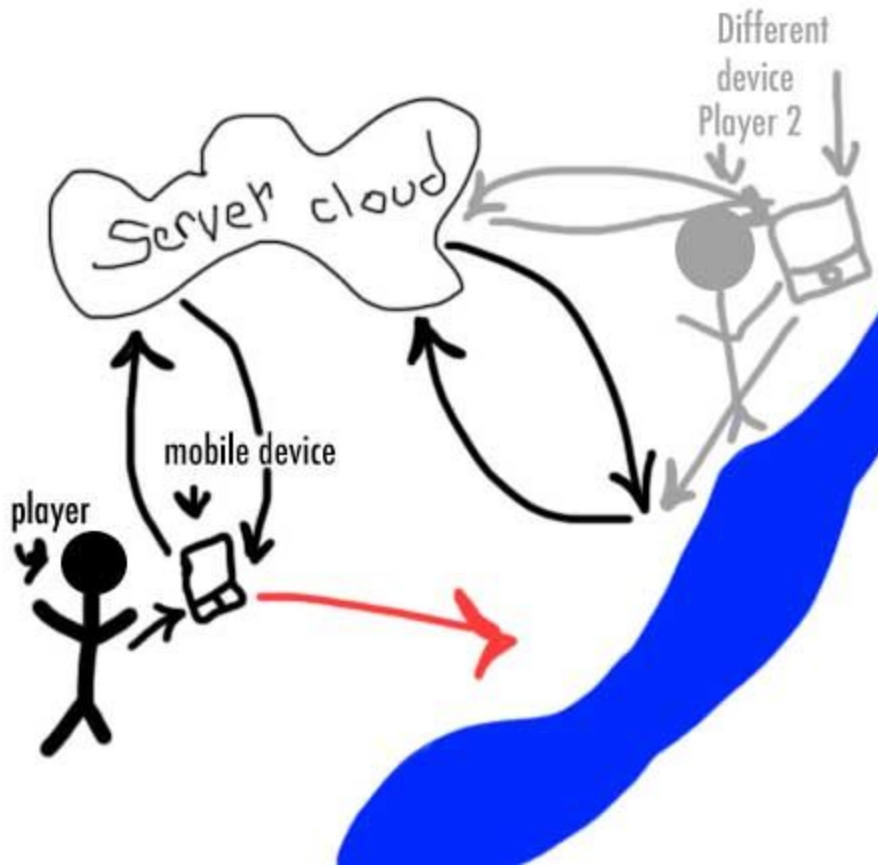


Figure 2 - Implementation Environment Diagram

The environment **Pandemic** will run in will be a mobile operating system. **Pandemic** will be supported on both IOS and Android operating systems. The game will be hardware agnostic as long as the hardware is running the last two versions of the operating system. This game should be able to run ideally on multiple resolutions, aspect ratios, and screen sizes.

While the **Pandemic** front-end will run on multiple mobile platforms, the pandemic back end will run on an AWS server in the cloud. This will allow seamless integration between multiple hardware devices as long as they share the same protocol. In the above diagram, a player with an Android device is communicating with the server to deploy his strain into a water source. An opposing faction's player on an IOS device is communicating with the same server to deploy their strain into the same body of water. The server is mediating the two and allowing concurrent gameplay, as well as scanning location data to verify that the client is telling it it has deployed to a body of water.

## Partner or Collaborative Applications

N/A

## Off-the-Shelf Software

Google location services

We won't be reinventing the wheel. **Pandemic** will use Google's existing database of maps and points of interest and dynamically allocating our own information to populate parts of it. Google offers a developer toolkit to integrate their location services which **Pandemic** can make full use of. We will use the business information from Google's Places API to identify between parks, lakes, grocery stores, hubs, etc.

<https://developers.google.com/places/>

Amazon Web Services

We believe that due to budget and lack of existing infrastructure, it makes sense for **Pandemic** to rent servers from Amazon rather than building our own farm. We've seen AWS host other services with high user demand such as Netflix streaming and feel that it will be sufficient to host **Pandemic**.

IOS / Android

To optimize performance of **Pandemic** it will need to be developed natively on each platform rather than on a cross-platform system. We will need to develop directly on IOS / Android and integrate the provided APIs to interface with the phone's hardware to drive retrieval of location data.

## Anticipated Workplace Environment

**Pandemic** will be played in any environment, including indoors and outdoors, standing or sitting, in heat or cold, light or dark, thus the game must be tuned to accommodate the more extreme conditions that may affect play. For example, the

game is to be playable with sound turned off. This is both because we want players to be able to have the full gaming experience even if they're somewhere where they must be quiet, or if they're somewhere that's too noisy to hear the game.

Another important thing to consider is that users will often be mobile while playing the game, meaning they might need to divert their attention off screen to watch where they're walking. A huge problem with *Pokemon Go* is that users became too distracted playing the game because they didn't want to miss a Pokemon. *Pandemic* will overcome this by allowing the user to delay certain actions until it is appropriate to complete them. For example, the user will be allowed to delay the start of a battle between an opposing faction's vaccine/virus in order to reach an area in the real world that is safe. This gives the user a couple of minutes to relocate themselves so that they're not battling viruses/vaccines in the middle of a busy street.

There will also be notifications that are sent to players for certain events like passing a member of your faction. This way, users don't need to constantly have the application open. However, there will always be things that the user isn't notified about, like contracting a virus/vaccine that an opposing faction has hidden.

### **Schedule Constraints**

The game must be released before its software becomes obsolete or incompatible with newer devices, otherwise the game will be undesirable to many users. Also, as *Pokemon Go* fandom dies down, it's important for *Pandemic* to secure its place as the next popular location-based game. Missing this window of opportunity may result in a less than adequate return on investment for the stakeholders. The main stakeholder, John T. Bell, therefore mandates that the project be completed by the end of May 2017.

## **Budget Constraints**

**Pandemic** will require a budget of 4.20 million dollars to be developed. This budget includes hiring developers, testers, BA's and PM's. It also includes the cost to rent Amazon AWS services and keep them spinning until the game builds up a user base and goes into the black. The budget will also be used to purchase dev machines and test machines. We will need access to the last two generations of Iphone, Ipad, and Android devices with screen sizes ranging from 4+ inches across multiple OS builds to ensure maximum compatibility and reach.

## **Naming Conventions and Definitions**

### **Definitions of Key Terms**

**BioHydra:** One of the two teams that compose Pandemic. This is the team focused on production and distribution of viruses.

**VacSect:** One of the two teams that compose Pandemic. This is the team focused on production and distribution of vaccines, and by extension is responsible for stopping the viruses spread by BioHydra.

**BH:** Acronym for BioHydra.

**VS:** Acronym for VacSect.

**Faction:** Can refer to either BioHydra or VacSect - simply another word for team.

**Virus:** The method of attack used by BH. It refers to a medical virus, such as the common cold, but used as a fictional creation of the faction.

**Vaccine:** The method of defense/attack used by VS. It refers to a medical vaccine, but used as a fictional creation of the faction.

**Battle:** The minigame played when receiving a vaccine or virus from an enemy player. It is composed of a Battleship-style board game wherein the player attacks unknown tiles and seeks to destroy all of their opponent's tiles (similar to ships in Battleship).

**Combat:** See "Battle"

**Bio Bomb:** An attack used by BH which designates a real-world area as an Infected Zone.

**Infected Zone:** A real-world region which passes along a chosen virus to any players that pass through the area.

Vac Bomb: An attack used by VS which designates a real-world area as a Vaccinated Zone, and clears Infected Zones.

Vaccinated Zone: A real-world region which passes along a chosen vaccine to any players that pass through the area.

## **UML and Other Notation Used in This Document**

This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in [4]. Any exceptions are noted where used.

## **Data Dictionary for Any Included Models**

N/A

## **Relevant Facts and Assumptions**

### **Facts**

The game will make use of real-world location, requiring users to provide that information.

### **Assumptions**

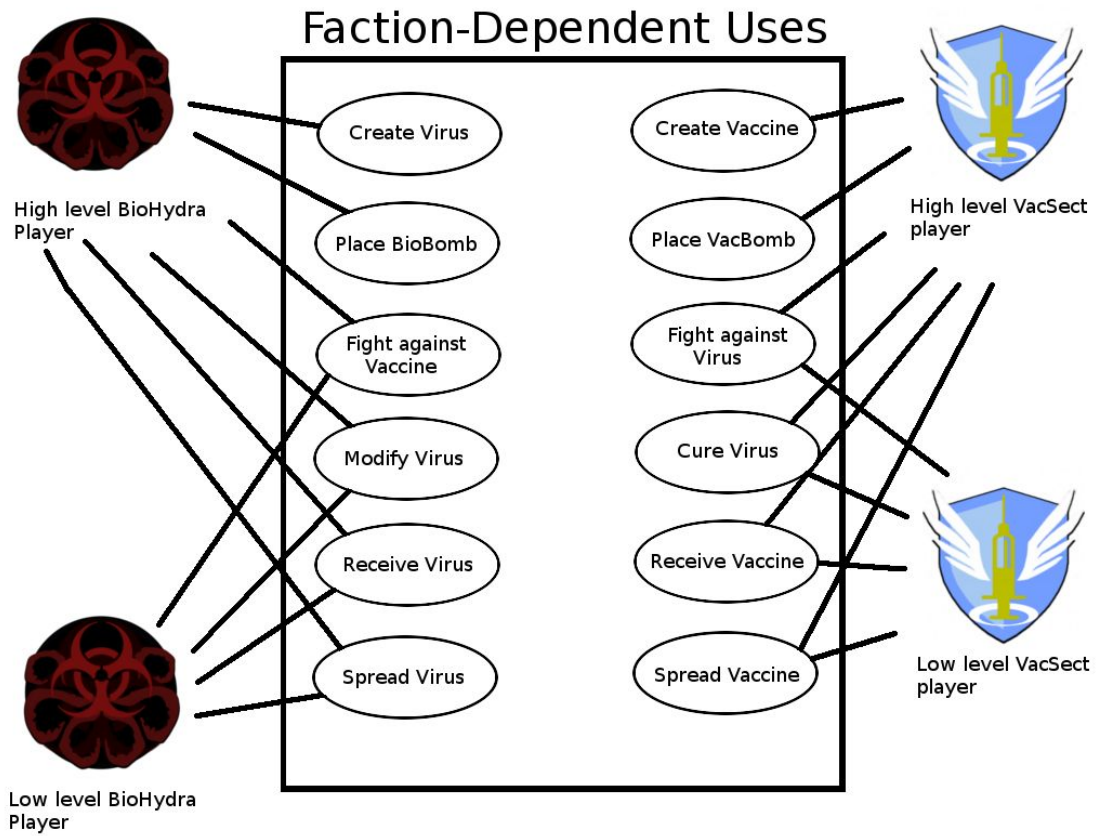
There are external services that can provide a map of the world to use in the game.

Real-world location can be tracked on the device being used.

# Requirements

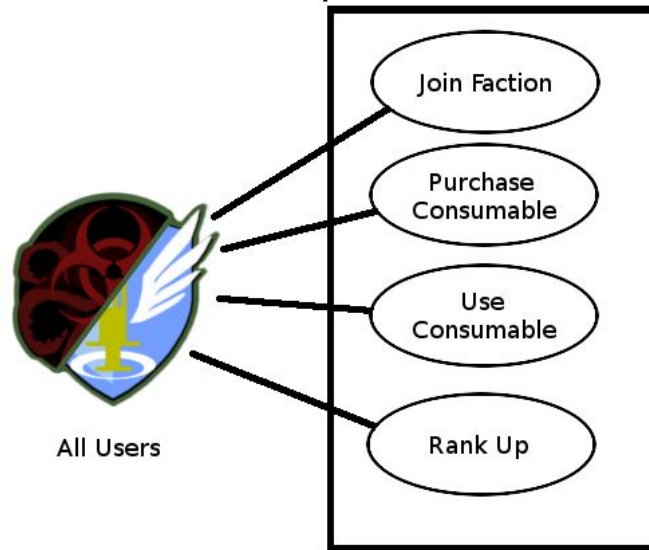
## Product Use Cases

### Use Case Diagrams





## Faction-Independent Uses



### Product Use Case List

1. **Create account/ Join faction**
2. **Create Virus**
3. **Create Vaccine**
4. **Fight Virus**
5. **Fight Vaccine**
6. **Receive Virus**
7. **Receive Vaccine**
8. **Spread Virus**
9. **Spread Vaccine**
10. **Modify Virus**
11. **Cure Virus**
12. **Place BioBomb**
13. **Place VacBomb**
14. Use Consumable
15. Purchase Consumable
16. **Rank Up**

## Individual Product Use Cases

<i>Use Case Name</i>	Create Account / Join Faction
<i>Participating Actors</i>	Initiated by active Client program Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. Client establishes session with server</li> <li>2. Client inputs information and sends protocol to server.</li> <li>3. Server checks information and returns whether username and email are available or already exist.</li> <li>4. If &gt;0, Client accepts new account and prompts user with the welcome screen.</li> <li>5. Client prompts user to select one of the two rival factions in game.</li> <li>6. Client sends a packet to server with the user's faction choice, and their database entry is updated.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User has installed Pandemic and is launching it for the first time.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● A new user account at level one successfully begins a game.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>● A new account can be created with ease.</li> <li>● Joining a faction and launching gameplay should be instantaneous.</li> </ul>

<i>Use Case Name</i>	SpreadStrains
<i>Participating Actors</i>	Initiated by active Client program Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. Client establishes session with server</li> <li>2. Client pings server for nearby users</li> <li>3. Server checks nearby area for users and returns the number of users found</li> <li>4. If &gt;0, Client sends list of viruses and vaccines to server</li> <li>5. Server sends list to each user in the area</li> <li>6. Each recipient user is updated with the new viruses/vaccines</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User is logged in and possesses at least one virus or vaccine</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● Virus is spread to nearby users</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>● Virus is propagated within 10 minutes</li> <li>● Receiving users are prompted to fight off the virus</li> </ul>

<i>Use Case Name</i>	Fight vaccine / virus
<i>Participating Actors</i>	Initiated by active Client program Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. User receives notification of having received a new virus/vaccine</li> <li>2. The user is prompted to enter into the fight off minigame or accept the strain passively as a cooperative member of the same faction.</li> <li>3. The complexity of the minigame is determined by the rank of the player, their items, and the complexity of the strain.</li> <li>4. The user either wins the mini-game and rejects the strain or loses it and becomes an unwilling agent of the opposing faction.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• The user must have received a new virus/vaccine</li> <li>• The user must be walking around with their location services turned on.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• User gains XP by successfully defending against a strain or becomes an unwilling carrier.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• Fights should prompt every time a user enters a hot zone.</li> <li>• Hot zones should have a transmission timer so that driving through doesn't immediately trigger 10 fights.</li> </ul>

<i>Use Case Name</i>	Accept Virus/ Vaccine
<i>Participating Actors</i>	Initiated by active Client program Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. User receives notification of having received a new virus/vaccine</li> <li>2. The client recognizes that the virus/vaccine is from the same faction and prompts the user to become a willing carrier.</li> <li>3. The user accepts and becomes an active host.</li> <li>4. Server is updated with the new data</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is playing and part of the same faction as the hot zone the user has entered.</li> <li>• User has received a new virus/vaccine</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• A user becomes an active host of a strain.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• Client should recognize the user's faction and incoming strain immediately</li> <li>• Client should not trigger fight minigame for a willing carrier.</li> </ul>

<i>Use Case Name</i>	Cure virus
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. The VacSect user enters a viral hot zone.</li> <li>2. The client pings the server with the user's coordinates and faction.</li> <li>3. The server sends the client a packet when the coordinates enter the range of the hot zone for a specified amount of time.</li> <li>4. The virus triggers the fight minigame with the VacSect user.</li> <li>5. The user wins the minigame and is rewarded with a boost item to create a vaccine to cure that strain.</li> <li>6. The VacSect user deploys the cure to the hot zone.</li> <li>7. The client pings the server to modify the hot zone.</li> <li>8. The hot zone is transformed into a quarantine zone and captured by VacSect.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● The user must be a member of VacSect</li> <li>● The user must actively enter a hot zone or come into contact with a virus.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● The hot zone becomes a quarantine area that is reflected for all users.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>● The hot zone to quarantine change should be immediate for all users.</li> <li>● User's entering the hot zone now should become immunized instead of infected.</li> <li>● There should be no cross-delay where users are still becoming infected.</li> </ul>

<i>Use Case Name</i>	Place BioBomb
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. A BioHydra member enters an area.</li> <li>2. The client pings the server with the user coordinates.</li> <li>3. The server pings the client with the landscape information that the client is around.</li> <li>4. The user enters his inventory and selects a BioBomb.</li> <li>5. The client displays a map of the area along with points of interest that will boost the deployment.</li> <li>6. The user selects a point on the map within range to deploy the bomb.</li> <li>7. The bomb is deployed to those coordinates and immediately creates a large radiating hot zone.</li> <li>8. The client sends a packet to the server with coordinates to change this location into a hot zone.</li> <li>9. Other Pandemic players in the hot zone are immediately prompted to fight or accent the strain.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● A user is a member of BioHydra.</li> </ul>

	<ul style="list-style-type: none"> <li>• A user has a BioBomb in the inventory.</li> <li>• A user is walking around with location data turned on and enters a location where the bomb is deployable.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• A radial area of the deployment becomes a hot zone.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• The map view should immediately change to show the hot zone.</li> <li>• Everyone with recent location in the new hot zone should be prompted without lag or delay.</li> </ul>

<i>Use Case Name</i>	Place VacBomb
<i>Participating Actors</i>	Initiated by active Client program Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. A VacSect member enters an area.</li> <li>2. The client pings the server with the user coordinates.</li> <li>3. The server pings the client with the landscape information that the client is around.</li> <li>4. The user enters his inventory and selects a VacBomb.</li> <li>5. The client displays a map of the area along with points of interest that will boost the deployment.</li> <li>6. The user selects a point on the map within range to deploy the bomb.</li> <li>7. The bomb is deployed to those coordinates and immediately creates a large radiating quarantine area.</li> <li>8. The client sends a packet to the server with coordinates to change this location into a quarantine.</li> <li>9. If the area was already a viral hot zone then the BioBomb destroys the hot zone.</li> <li>10. Other Pandemic players in the quarantine area are all either cured or immunized from that strain.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is logged in and possesses a VacBomb in their inventory.</li> <li>• The User is a member of VacSect</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• Deployed area is turned into a large quarantine.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• Area immediately changes.</li> <li>• There is no lag between bomb deployment, map change, and user immunization.</li> </ul>

<i>Use Case Name</i>	Create Vaccine
<i>Participating Actors</i>	Initiated by User Communicates with Server

<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. User chooses to create a new vaccine via the UI</li> <li>2. User gives the vaccine a name and sets up the vaccine's minigame board via the UI</li> <li>3. User submits the completed vaccine to the server</li> <li>4. Server validates/salts the input and filters any profanity in the name</li> <li>5. Server stores the new vaccine in the database</li> <li>6. Server adds vaccine to user's list of carried vaccines</li> <li>7. User is given a popup notifying them of successful creation</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User is logged in and meets the faction rank requirement</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● Vaccine has been created</li> <li>● The user possesses the new vaccine.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>● Vaccine is propagated within 10 minutes</li> <li>● Receiving users are prompted to fight off the vaccine or accept it.</li> </ul>

<i>Use Case Name</i>	Create Vaccine
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. User chooses to create a new vaccine via the UI</li> <li>2. User gives the vaccine a name and sets up the vaccine's minigame board via the UI</li> <li>3. User submits the completed vaccine to the server</li> <li>4. Server validates/salts the input and filters any profanity in the name</li> <li>5. Server stores the new vaccine in the database</li> <li>6. Server adds vaccine to user's list of carried vaccines</li> <li>7. User is given a popup notifying them of successful creation</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User is logged in and meets the faction rank requirement</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● Vaccine has been created</li> <li>● The user possesses the new vaccine.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>● Vaccine is propagated within 10 minutes</li> <li>● Receiving users are prompted to fight off the vaccine or accept it.</li> </ul>

<i>Use Case Name</i>	Create Virus
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. User chooses to create a new virus via the UI</li> <li>2. User gives the virus a name and sets up the virus's minigame board via</li> </ol>

	<p>the UI</p> <ol style="list-style-type: none"> <li>3. User submits the completed virus to the server</li> <li>4. Server validates/salts the input and filters any profanity in the name</li> <li>5. Server stores the new virus in the database</li> <li>6. Server adds virus to user's list of carried viruses</li> <li>7. User is given a popup notifying them of successful creation</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User is logged in and meets the faction rank requirement</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● A new virus has been created</li> <li>● The user possesses the new virus</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>● Virus is propagated within 10 minutes</li> <li>● Receiving users are prompted to fight off the virus</li> </ul>

<i>Use Case Name</i>	Use Consumable
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. User opens Pandemic and views the inventory screen.</li> <li>2. The client alerts the server to pull DB information and populate the account's inventory.</li> <li>3. The client displays a list of all of the available consumables.</li> <li>4. The user selects one to use.</li> <li>5. The client prompts the user with the ways that the user can deploy the consumable.</li> <li>6. ex) The user selects the consumable "Roadkill" and has option to reverse engineer it to create a low level strain.</li> <li>7. The client prompts them to the modify virus screen.</li> <li>8. On completion, the client sends a packet to the server to update the inventory, +1 strain and -1 roadkill.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User is logged in and has a consumable in the inventory</li> <li>● User is at a high enough level to use the consumable.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● A new strain has been created</li> <li>● The user possesses the new strain</li> <li>● The user no longer posses the consumable.</li> <li>● The user gains XP</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>● Duplicate consumables should stack on top of eachother.</li> <li>● Consumables should immediately disappear on use.</li> <li>● Consumables should not disappear if the next use case is cancelled.</li> </ul>

<i>Use Case Name</i>	Purchase Consumable
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. The user opens the game menu and navigates to the Pandemic store.</li> <li>2. The client pings the server and pulls a list of available items, their level requirements, and their cost in virtual Pandemic dollars.</li> <li>3. The client also displays the user's virtual wallet.</li> <li>4. The user finds an item that the user can afford and hits purchase.</li> <li>5. The client prompts the user to confirm the purchase.</li> <li>6. The client sends a packet to the server with an updated balance of the user's wallet, as well as an addition to their inventory.</li> <li>7. The user opens the menu and navigates to the Pandemic inventory.</li> <li>8. The consumable appears in the inventory.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User has is a high enough level to purchase the desired consumable.</li> <li>• User has enough currency to afford the consumable.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• Used currency is subtracted from the user's wallet.</li> <li>• New consumable is added to the user inventory.</li> </ul>
<i>Quality Requirements</i>	<ul style="list-style-type: none"> <li>• User can only purchase items that are below their rank.</li> <li>• User can only purchase items they can afford.</li> <li>• The wallet shall not allow for negative balance.</li> <li>• Accurate subtractions shall happen 100% of the time.</li> <li>• The new item shall appear in the inventory only once, but shall appear without fail 100% of the time.</li> </ul>

<i>Use Case Name</i>	Rank Up
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. An existing faction member performs a game task.</li> <li>2. Game tasks include any of the above listed requirements.</li> <li>3. On completion of the game task, the user sees the XP bar increase.</li> <li>4. The client sends the server the user's total XP and updates the DB.</li> <li>5. Once the amount of XP gained from completing a task becomes greater than the set amount needed per level, a rank up is triggered.</li> <li>6. The client prompts the user to explore the features unlocked in the new rank.</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is logged in and a faction member</li> <li>• User has gained enough XP to rank up</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• User rank increases by 1.</li> </ul>



*Quality Requirements*

- XP should be fairly scaled and distributed.
- Every level increase should also increase the amount of XP required to rank up again.

## Functional Requirements

Requirement # 1	Use Case # 1
Description: The two factions, BioHydra and VacSect, shall remain balanced in terms of number of players per faction.	
Rationale: If one faction has significantly more players than the other, it will have an unfair advantage over the other.	
Fit Criterion: Each faction shall not exceed 58% of the total number of players. ex: $(\# \text{ in VacSect})/(\text{total}) \leq 0.58$ AND $(\# \text{ in BioHydra})/(\text{total}) \leq 0.58$	
Customer Satisfaction: 8/10	Customer Dissatisfaction: 8/10
Priority: High	

Requirement # 2	Use Case # 2
Description: High level members of BioHydra must be able to create viruses.	
Rationale: Without the creation of viruses, the game will not have any progression.	
Fit Criterion: BioHydra players above a required level shall be able to create their own viruses ex: $(\text{Player Level}) > 50$ AND $(\text{Player Faction}) = \text{BioHydra} \rightarrow \text{Player can create virus}$	
Customer Satisfaction: 5/10	Customer Dissatisfaction: 10/10
Priority: High	

Requirement # 3	Use Case # 3
Description: High level members of VacSect must be able to create vaccines.	

Rationale: Without the creation of vaccines, VacSect will not be able to compete with BioHydra.	
Fit Criterion: BioHydra players above a required level shall be able to create vaccines for enemy viruses. ex: (Player Level) > 50 AND (Player Faction) = VacSect → Player can create vaccine	
Customer Satisfaction: 3/10	Customer Dissatisfaction: 10/10
Priority: High	

Requirement # 4	Use Cases # 4, 5, 6, 7
Description: Players must be able to fight against enemy viruses/vaccines	
Rationale: Without this interaction, the game will lack substance.	
Fit Criterion: If a user is within X meters of enemy player carrying a new virus, a fight is triggered ex: DistanceBetween(Player.Location, Enemy.Location) <= X → Fight triggered	
Customer Satisfaction: 8/10	Customer Dissatisfaction: 10/10
Priority: High	

Requirement # 5	Use Case # 16
Description: Players must be able to Rank up	
Rationale: Users should be rewarded for playing and being successful within the game	
Fit Criterion: UserLevel = (UserExperiencePoints /100)	
Customer Satisfaction: 7/10	Customer Dissatisfaction: 4/10
Priority: Medium	

Requirement # 6	Use Cases # 12,13
-----------------	-------------------

Description: The server must store the location of BioBombs and VacBombs	
Rationale: System must know when a user is in range of a BioBomb or VacBomb	
Fit Criterion: When bomb is placed; BioBombLocation = User.currentLocation	
Customer Satisfaction: 7/10	Customer Dissatisfaction: 8/10
Priority: High	

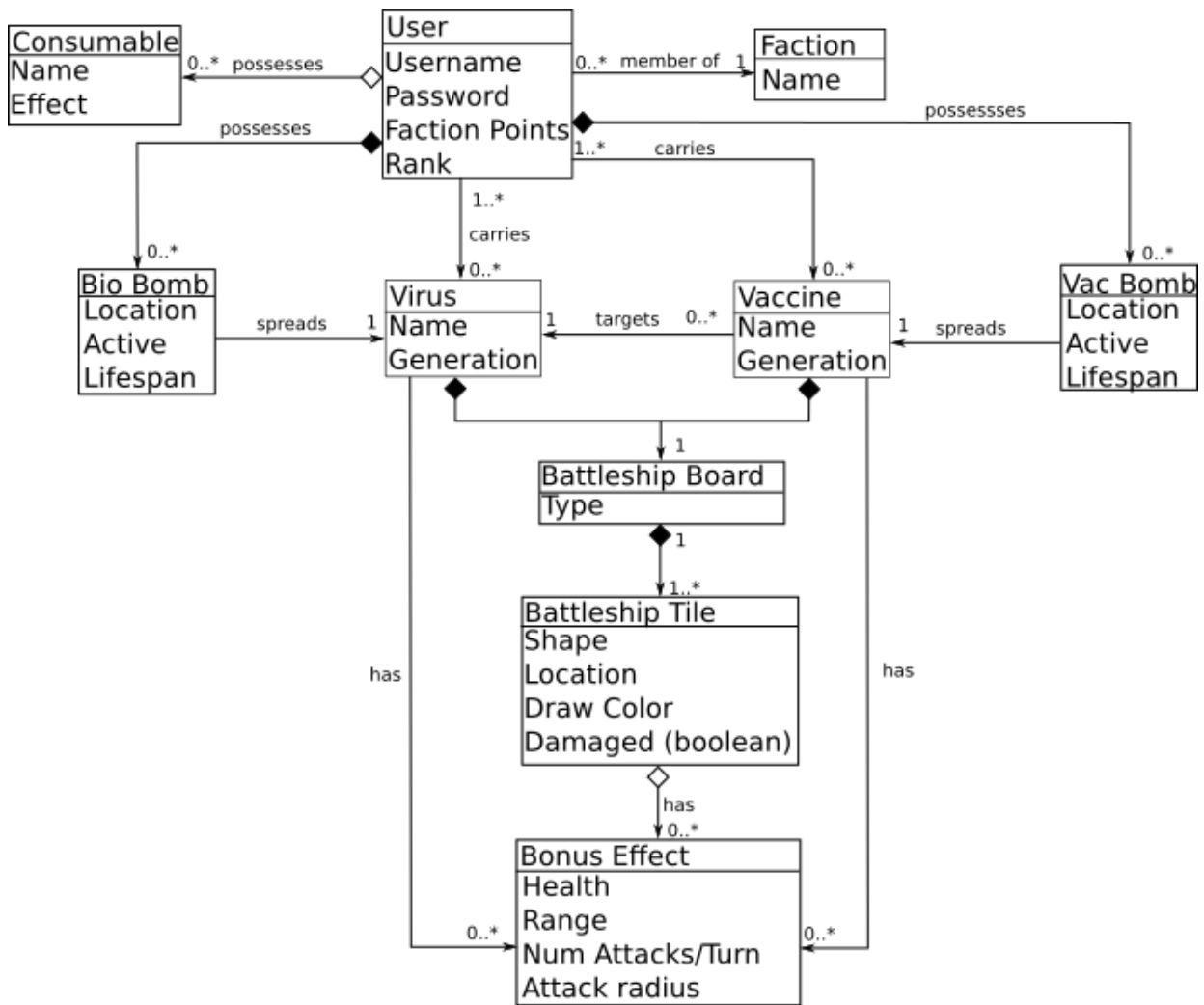
Requirement # 7	Use Cases # 8, 9
Description: Players must be able to spread their virus/vaccine through use of items and by getting close to them in the real world	
Rationale: Players need to be able to complete the main objective of the game.	
Fit Criterion: If a user is within X meters of potential recipient or their item is in effective range ex: DistanceBetween(Player.Location, Enemy.Location) <= X OR Player.Item.canInfect == true → Virus/Vaccine will spread	
Customer Satisfaction: 8/10	Customer Dissatisfaction: 10/10
Priority: High	

Requirement # 8	Use Cases # 11
Description: The system must keep track of which Virus a Vaccine will cure	
Rationale: BioHydra will dominate if their viruses are indestructable	
Fit Criterion: BioHydraPlayer.Rank < VacSectPlayer.Rank AND VacSectPlayer.Cures(BioHydraPlayer.Virus) → Virus can be destroyed	
Customer Satisfaction: 8/10	Customer Dissatisfaction: 10/10
Priority: High	

Requirement # 9	Use Cases # 14,15
Description: The system must contain a store where players can purchase consumables using real currency or experience points	
Rationale: Players need to get items somehow	
Fit Criterion: All items will be available in the store and purchasable with a variable number of experience points	
Customer Satisfaction: 6/10	Customer Dissatisfaction: 3/10
Priority: Medium	

## Data Requirements

Class Model:



## Performance Requirements

### Speed and Latency Requirements

- Location-based events should initiate within 10 seconds of entering the effective zone. E.g a user steps into range of a BioBomb.
- User's location should be tracked within 30 seconds of their movement.
- The app should launch and load to the login screen within 4 seconds.

### Precision or Accuracy Requirements

- All locations must be accurate within  $\pm 5$  feet.

## **Capacity Requirements**

- The product shall support simultaneous play of 5,000 players per server region within the period of 9:00 am to 11:00 pm. Maximum load during other periods will be 1500 simultaneous users.
- The product, upon release, shall support the storage of login information of 100,000 users across the globe.
- The product shall support the storage of up to 4000 different viruses, including their board and subsequent modifications, as well as any vaccines created for them.

## **Dependability Requirements**

### **Reliability Requirements**

- Pandemic servers shall maintain 99% uptime.
- In the event of a new or replaced local device, users should be able to log in and recover their progress.

### **Availability Requirements**

- The product shall achieve 96 percent uptime.
- The product shall make increased efforts to remain available 100 percent of the time during expected high traffic periods of time, such as the first few weeks of product launch, during holiday breaks, after extensive media coverage, etc.

### **Robustness or Fault-Tolerance Requirements**

- In the event of connection loss, the game shall locally save any information meant for the central server. Upon reconnection, the uncommunicated data is resent automatically.
- The game shall not allow the user to continue to play after network connection has been lost.

### **Safety-Critical Requirements**

- The game shall warn users not to loiter in unacceptable areas (in the airport, on private property).
- The game shall remind users to pay attention to their surroundings on a regular basis.

## **Maintainability and Supportability Requirements**

### **Maintenance Requirements**

- The product shall not be down for maintenance, except for complete server failure, for longer than 3 hours per 1-week period.
- Scheduled maintenance should occur between 1:00 am and 4:00 am in each region.

### **Supportability Requirements**

- The game's features should be explained through brief introductory videos for each feature.
- Unexpected exceptions in the game should generate logs and crash reports that are sent to Pandem Inc.
- Multi-regional support technicians should be staffed.
- User's should be able to call in and reach a support technician between 9:00am and 5:00pm on weekdays.
- A forum and FAQ section will exist in the Pandemic website.
- Someone from the support staff should moderate the forum.

### **Adaptability Requirements**

- Pandemic must run on devices with GPS and location data turned on.
- The game must run on IOS and Android devices with screens ranging from 4 inches and up in multiple aspect ratios.
- The game is designed to be played on the move.
- The game will be rolled out globally.
- Language localizations should exist for every region.

### **Scalability or Extensibility Requirements**

- Pandemic should be globally scalable.
- The game should be able to support 1 million active users at launch.
- The game should be able to support 3 million active users in the first year.
- The game should be able to track the locations of all users without overloading.
- The game should be able to maintain all infected strains and locations without needing to overwrite or drop any data.
- The game should allow for multiple active strains per active user consistently.

### **Longevity Requirements**

- Release 1 should be able to run consistently for a full year.

- Server side elements should be able to run for at least 3 years for upcoming releases.
- The release should be able to run on all new and upcoming devices within that release timeline.

## **Security Requirements**

### **Access Requirements**

- Users shall not be able to see the location of any other users.
- Users shall not be able to identify any other users.
- Only logged-in users shall be able to play the game.

### **Integrity Requirements**

- The system shall CAPTCHA-protect account creation.
- The system shall not allow data entry from non-users.
- The system shall validate all input.

### **Privacy Requirements**

- The database shall not store the identifying information of a user linked to their location
- The database shall only store the locations of Bio- and Vac-Bombs, usernames, and virus data
- The database shall not store any personal information not expressly entered by the user



## **Audit Requirements**

- Pandemic should store only the usernames and passwords of users, as well as their chosen faction and the viruses/vaccines they carry.
- The game should keep anonymous location maps to track movement and hotspots.

## **Immunity Requirements**

- Security updates will be pushed out periodically to defend against exploits.
- Servers will be professionally maintained and inaccessible to users.
- Accounts found with exploits or corrupt APKs shall be banned.
- The game executable will only be available from the Google and Apple store.

## **Usability and Humanity Requirements**

### **Ease of Use Requirements**

- The player should be able to get in and playing in less than 10 minutes.
  - Once an account is created, Interactive tutorials will run to explain the setup and startup of the game. No more than 3 tutorials will run at a time in order to keep the game flow consistent. Features are introduced with tutorials as they are unlocked, so the player is learning the rules as s/he progresses, without feeling overwhelmed. With this progression format, at no point will be player struggle for longer than 10 minutes to figure out any aspect of gameplay.
- Casual players should be able to pick up where they left off.
  - The previous tutorials will be stored in an archive that can be replayed.
  - A Pandemic Wiki and FAQ section will also be made available to get players up to speed.
- The user should commit very few user errors.
  - The on rails approach of the first couple of level ups and feature introductions will ensure that beginners do not make a variety of user errors.
  - Confirmation windows will also exist for major game features to ensure that users aren't deploying a strain or using a consumable unintentionally.
  - Pandem Inc. will issue users surveys to describe any committed user errors periodically. This information will be gathered by the UX experts and used to improve the UI in the next release.
- Users shall be able to provide reviews and ratings through their respective app store to provide satisfaction data and feedback to Pandem Inc.

- A support technician will monitor user feedback and star ratings to suggest improvements to Pandemic to address complaints.
- New features and tweaks will be rolled out periodically to constantly improve gameplay.

## **Personalization and Internationalization Requirements**

- Users shall be able to choose the language of their choice.
- Users shall be able to create a custom username and avatar to represent them.
  - This avatar will be stored in their account preferences and identify them to the community at large.
- Users will be able to create their own signature strains, which will display their avatar to infected/vaccinated players and players crossing a hot zone/quarantine.
- Users will be able to create and join local factions of their team which can be customized with a logo and name.

## **Learning Requirements**

- Pandemic shall be inviting to new players and easy to learn.
  - By focusing on progressive difficulty ramping and guided tutorials, new players will never feel overwhelmed.
  - The length of guided tutorials ensures that players spend no more than 3 minutes per level up to learn new features.
- A high level player shall be able to perform advanced features with relative ease.
  - By the time a player reaches level 50, they will have finished every guided tutorial and spent countless hours earning the rank of master. At this point, over 200 hours of gameplay time and a new feature allow the player to produce unique strains from scratch with ease.
- A skilled player that hasn't logged in in a while will be able to quickly acquaint himself with Pandemic.
  - A Pandemic Wiki, FAQ and archived tutorials will be accessible to all players.

## **Understandability and Politeness Requirements**

- New users must watch interactive tutorial on how to play the game.

- Users shall not be exposed to the underlying implementation of the game, especially in regards to the server connection. No language involving servers will be displayed to users. Generic error messages will be displayed instead of specific technical issues. Example messages include “Network connection error” or “We’re working on fixing a problem now. Please try again later”.

### **Accessibility Requirements**

- The product shall be made usable by deaf people through the use of vibrating notifications.
- The product shall be usable by color blind people through the use of distinctly shaped team logos, as well as a Colorblind mode for the user interface, which features a more contrasted color scheme

### **User Documentation Requirements**

- Instructional how-to-play documentation, which is displayed as part of the interactive tutorial, will be made available to the user within the app for viewing at a later date.
- Any large updates to the application will be accompanied by an update to the instructional documentation, if deemed necessary by the developers. A technical writer will receive direction from the developers on what has changed.
- A list of all legal warnings included in the product (including those involving privacy of data, safety warning about moving while playing, and legal warning about the banning of the game in certain areas) will be made available to the user through a link in the app which redirects the user to **Pandemic**’s web page.
- 

### **Training Requirements**

- No training beyond the initial interactive tutorial shall be required for a user to be able to use the product to its full potential.

### **Look and Feel Requirements**

#### **Appearance Requirements**

The product shall use two color schemes, one for each faction. BioHydra shall have a red theme, and VacSect shall have a light blue theme.

The product’s interface shall be attractive to a young adult audience. A representative sample of young adults must not have any major concerns with the UI.

The product shall ensure that all text is easily readable. Font sizes smaller than 8 are not allowed.

## **Style Requirements**

The product shall have a “benign” feel to it. Users shall not express concerns about actually influencing the real world.

## **Operational and Environmental Requirements**

### **Expected Physical Environment**

- As many events will occur while the user is not actively paying attention to their phone, **Pandemic** will notify users through both noise and vibration notifications. Users can change the settings for these notifications if they do not wish to be disturbed.
- The product shall remind users to remain aware of their surroundings, especially if it detects that the user is moving while interacting with the game. These reminders shall occur at every startup and once every 10 times the app has detected movement while playing.

### **Requirements for Interfacing with Adjacent Systems**

- Pandemic shall work on the last two operating systems of Android and iOS across both phones and Tablets. Pandemic shall also work on any OS updates that are being released the year of Pandemic’s first release.
  - The client side code for Pandemic will be abstracted and hardware agnostic. The client side code should simply be a front end that communicates with a specified and agreed upon protocol to the server. The front end will support resizing and multiple resolutions. This will allow Pandemic to be cross platform, centralized, and controlled across all devices. This will also ensure that all users have the same Pandemic experience regardless of their hardware needs or limitations.
- Pandemic servers shall integrate with Google places API regardless of version or update.
  - Pandemic servers will be maintained to always be compatible with the Google maps and places APIs. Pandemic will be able to send the raw device coordinates to Google and pull back points of interest and relevant

location data. The servers should interface to seamlessly make these data exchanges millions of times a day.

## **Productization Requirements**

- The product shall be available for download in the iOS app store and the Google Play Store.

## **Release Requirements**

- **Pandemic** will deploy updates a minimum of every 3 months while the product maintains 500,000 users or more. Below this threshold, updates will be provided as needed, as determined by the development team.
- Server updates can happen as many times a day as necessary to ensure proper functionality of the application, provided the updates do not violate the previously outline availability requirements.
- In the event that a serious security issue is patched in an update, **Pandemic** will not allow users to connect to the servers unless they are using the latest version. Otherwise users will be able to continue using old versions of the application while being reminded to update at each launch.

## **Cultural and Political Requirements**

### **Cultural Requirements**

**Pandemic** is a global game, and as such shall be playable by anyone anywhere in the world.

- The product will display distances in whatever unit fits the standard system of the user's location. For examples, the Imperial system in the USA and Metric in many other places. All data stored in databases is stored as if the user were in the USA, and then converted to be localized when loaded from the database.
- Users will be able to choose their preferred localization settings, including language, and that preference will be satisfied regardless of location.
- Users will be able to request their preferred language if it is not already supported by default. By default, **Pandemic** will support English, Spanish, Mandarin, Russian, Portuguese, and Japanese. Any language that is requested by over 100,000 users will be added into the product within 1 month.

### **Political Requirements**

- Developers' opinions regarding app design shall hold equal weight with that of their superiors. Any dispute that cannot be resolved between a Developer and his superior shall be put to a majority vote of all Developers of the appropriate team and all managers. The CEO may serve as a tie-breaker.

- The CEO reserves the right to immediately resolve any democratic conflict among managers and developers with a solution of his choosing. This avoids wasting development time on arguments.

## **Legal Requirements**

### **Compliance Requirements**

- Personal information shall be implemented so as to comply with the Data Protection Act.
- The developing company shall accept requests from companies who wish to ban the game on their premises. For example, a school may have a problem with students playing the game on campus. All approved requests will be honored by the product.
- Pandemic is the intellectual property of the developing company and shall be protected as such by the legal department through patents and litigation against any infringing parties.

## Standards Requirements

- Pandemic shall comply with AAA game development standards.
- Pandemic shall comply with Pandem Inc coding style standards:
  - All code shall be indented with tabs instead of spaces.
  - All variables shall be named as two words with a lowercase letter followed by an uppercase letter. `//boolean isDamaged //String userName, passWord`
  - Development should be done in adherence to scrum methodology.
  - Sprints shall last 2 weeks.
  - Story points shall be considered points of effort that scale according to the Fibonacci sequence.
  - For every compilation failure, all developers must take a shot

## Design

### System Design

#### Design goals

- Pandemic should have as little server downtime as possible. Noting the negative reaction to Pokemon Go's constant server failures, this is one of the key design goal, especially as the system scales.
- The server response time should be as quick as possible. Players will constantly be polling databases for the locations of other users, BioBombs and VacBombs, and more. If these operations are slow, the game will slow to a crawl and users will be frustrated.
- The game must be simple to begin playing. New users should not require much effort to create an account, learn how to play, and begin having fun.

### Current Software Architecture

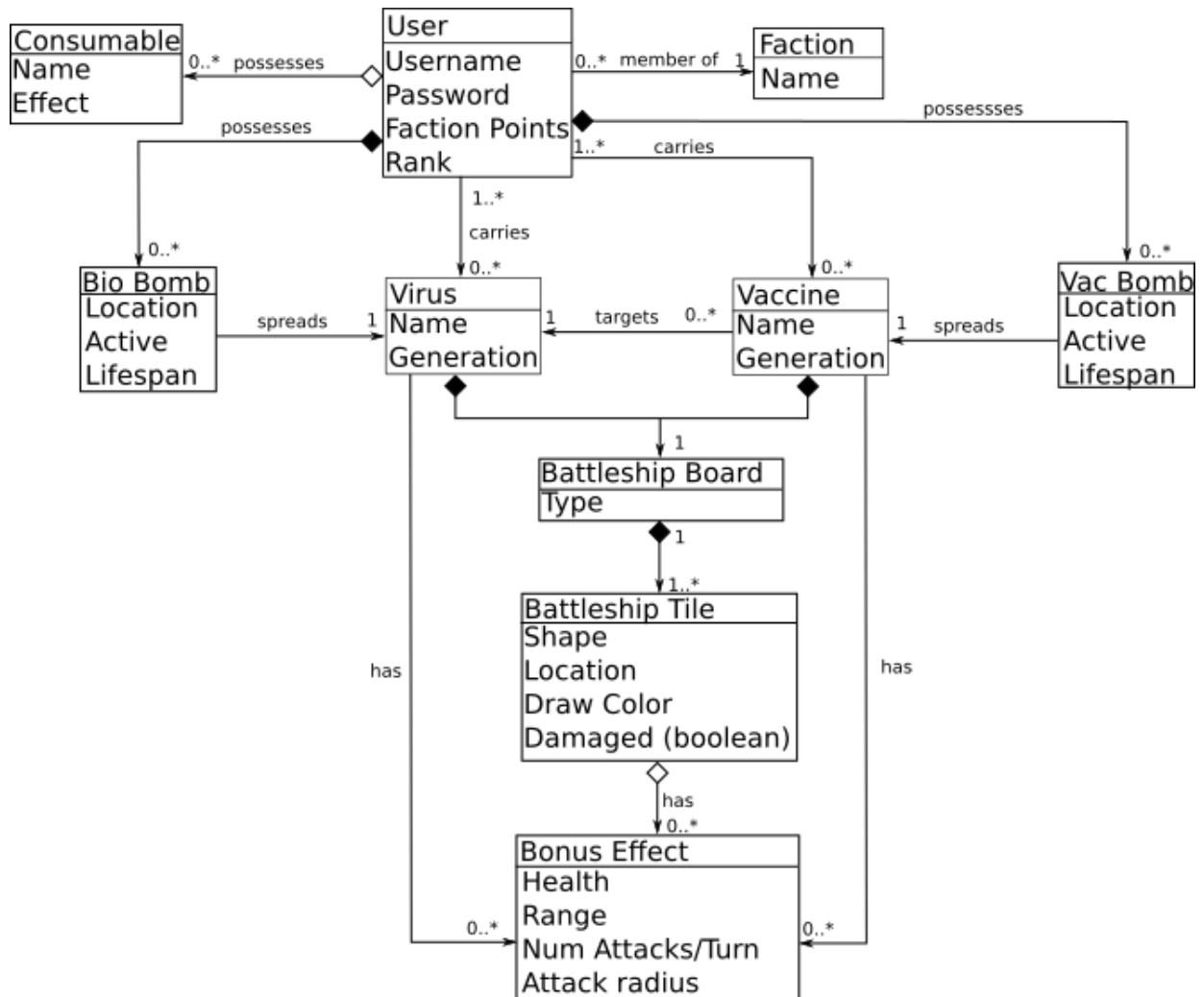
There is no existing software architecture that must be taken into account for the development of *Pandemic*. The mobile application does not interact with any other software on the device and the servers are not running any code that was not written specifically for this project.

### Proposed Software Architecture

#### Overview

*Pandemic* uses 2 main codebases to create the user experience: the mobile application which runs on each user's device and the server code which runs on many servers all throughout the world. The mobile application provides the user interface to players and communicates with the server. The server processes information from many different players and sends the result of this processing back to users. See [figure 1](#).

## Class Diagrams





## **Dynamic Model**

### **Subsystem Decomposition**

1. Mobile system
  - a. collect data from users
  - b. display data to users
2. Server system
  - a. store persistent data
  - b. keep track of users in an area
  - c. handle information sent from users
  - d. report important information back to users

### **Hardware / software mapping**

The mobile application will run on user's mobile devices, namely Android or iOS devices which have an internet connection.

The server software will run on many different servers throughout the world, namely rented Amazon AWS servers.

### **Data Dictionary**

The database will contain information regarding the following:

- user account
  - item list
  - username
  - password
  - level
  - faction
- BioBomb/VacBomb
  - location
  - owner
  - faction

### **Persistent Data management**

Most persistent data will be stored in database on the game's server. This include players login information and all information related to their account (player's level, number of items, etc.).

There will be minimal information storage on the mobile devices, such as storing a user's login information so they don't have to type their credentials every time they start the app. Persistent storage in the cloud is preferred because it frees up memory on the user's device for other things and prevent users from tampering with things. For example, if a user knew their list of items was stored on their phone somewhere, they could perhaps reverse engineer a way to give themselves more items and give themselves an unfair advantage.

### **Access control and security**

*Pandemic* players will only be able to begin playing after they have successfully logged in to an existing account or created a new one. The first time a user logs in from a device which is not the same device the account was created on, the user will receive an email notifying them of this activity so that they can confirm someone is not using their account.

Physical access control of databases is out of the company's hands, as we are renting servers from Amazon. However, it is important to ensure that each connection to the database comes from a legitimate user.

### **Global software control**

Server software can be globally controlled by the *Pandemic* developers through a Master server which can do things like push updates to all other servers, manage accounts of users, and change different parameters.

The configuration of the mobile software can be globally controlled by pushing updates to all servers which will send changes to any user when they connect. This will be done without the user's knowledge in the case that the change will not drastically affect them or take a large amount of time.

### **Boundary conditions**

- User participation reaches a number beyond the estimated and planned for numbers. This could cause server crashes, potentially resulting in system-wide failure for a particular region.
- The APIs that *Pandemic* relies on do not function properly. For example, location data may be spotty in certain areas, and if the Maps API doesn't report that to *Pandemic*, we will just see the data and treat as correct.
- The persistent data stored on the servers somehow becomes corrupt. This could result in users being served incorrect information about their accounts.

### **Subsystem services**

See section “Subsystem Decomposition”

## **User Interface**

The user interface of *Pandemic* exists on the mobile devices and will be similar to that found in Pokemon Go. The main screen is an augmented reality display where users can see the location of items like BioBombs and VacBombs. There will be separate screens for Virus/Vaccine design, the BattleShip-style mini game, buying items, and other account-related menus.

## **Object Design**

### **Object Design trade-offs**

The top-down organizational approach taken in the class design makes finding a particular virus’s effects difficult to navigate to. This tradeoff is managed by making it possible to navigate to them directly via the User class.

### **Interface Documentation guidelines**

All interfaces must be well documented with JavaDoc (or similar) comments throughout the development process. All classes and their methods should have descriptions above them. A documentation compiler (such as the JavaDoc compiler for Java code) will be run on the codebase daily so that any new changes will quickly be reflected in the API documentation.

### **Packages**

Developers must use their own discretion to choose which package to place their class in. No package can contain more than 40 classes. Packages will be evaluated and potentially reorganized every month.

### **Class Interfaces**

Developers must use their own discretion to decide when to define an Interface before developing its implementing Class. The general rule is if you can think of at least 3 “is-a” relationships between some classes, they should all implement the same interface.

## **Test Plans**

### **Features to be tested / not to be tested**

The product shall be tested to ensure that the following features are met:

1. Player can choose a faction upon account creation
2. Player can receive viruses and vaccines from players in their area
3. BioHydra players can modify their team's viruses
4. High level BioHydra players can create viruses and Bio Bombs, while low level players and all VacSect players cannot
5. High level VacSect players can create vaccines and Vac Bombs, while low level players and all BioHydra players cannot
6. Players can spread their viruses or vaccines passively
7. Players can fight against enemy viruses or vaccines that they have received
8. Players continue to carry viruses/vaccines if and only if they fail to fend them off
9. Servers can support 5000 simultaneous players during the specified hours.

### **Pass/Fail Criteria**

A test result will be considered a failure if they meet the corresponding criteria below:

1. Player cannot choose a faction upon account creation
2. Player does not receive a virus or vaccine when there is an enemy player within X meters of him/her
3. BioHydra player cannot modify a virus they have in their possession
4. High level BioHydra players cannot create viruses and Bio Bombs, or low level players or any VacSect players can
5. High level VacSect players cannot create vaccines and Vac Bombs, or low level players or any BioHydra players can
6. Players do not spread their viruses or vaccines passively
7. Players cannot fight against enemy viruses or vaccines that they have received
8. Players do not continue to carry viruses/vaccines if they fail to fend them off, or players continue to carry viruses/vaccines they have successfully beaten
9. Any server slows down beyond a comfortably playable level under a load of 5000 simulated players during the specified hours.

## Approach

Most tests will be carried out through playtesting - i.e. the tester will play the game with a relevant user account and check that the features are available to them.

We will also run automated tests on the servers to ensure their software is working properly.

## Suspension and resumption

- Automated server tests must run to completion to ensure their results are accurate.
- Human testers may pause and resume their manual testing at any point.

## Testing materials ( hardware / software requirements )

- Automated unit tests will take place on the Amazon AWS servers to ensure requirements are met.
- Human testers will use both Apple and Android mobile devices to test the functionality of all user-facing features. Tests will first be done using the latest technology (newest operating system and newest hardware). Then, some backwards compatibility testing will be done using phone hardware up to 3 generations old and operating systems as old as Pandemic's design supports.

## Test cases

<i>Test Case Name</i>	Create Account / Join Faction
<i>Test Steps</i>	<ol style="list-style-type: none"><li>1. Open the app as a user that is not logged in</li><li>2. Choose to create an account via the UI</li><li>3. Select a username and password</li><li>4. Click OK</li><li>5. User should be prompted to choose a faction</li><li>6. Select a faction</li><li>7. Click OK</li></ol>
<i>Entry condition</i>	<ul style="list-style-type: none"><li>• User has installed Pandemic and is launching it for the first time.</li></ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"><li>• A new user account at level one successfully begins a game.</li></ul>

<i>Use Case Name</i>	SpreadStrains
<i>Test Steps</i>	<ol style="list-style-type: none"><li>1. Log in as two enemy users on two separate devices, who each have a</li></ol>

	<ul style="list-style-type: none"> <li>virus or vaccine</li> <li>2. Move the devices within spread range</li> <li>3. Within ten minutes, each user should have a notification that they have received a new virus/vaccine</li> </ul>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User is logged in and possesses at least one virus or vaccine</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● Virus is spread to nearby users</li> </ul>

<i>Use Case Name</i>	Fight vaccine / virus
<i>Test Steps</i>	<ol style="list-style-type: none"> <li>1. Receive notification on device</li> <li>2. Click notification or open the app</li> <li>3. Should be presented with the incoming virus/vaccine</li> <li>4. Click “Fight” button</li> <li>5. Play minigame to completion</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● The user has received a new virus/vaccine</li> <li>● The user must be walking around with their location services turned on.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● User gains XP by successfully defending against a strain or becomes an unwilling carrier (and gains no XP).</li> </ul>

<i>Use Case Name</i>	Accept Virus/ Vaccine
<i>Test Steps</i>	<ol style="list-style-type: none"> <li>1. Receive notification on device</li> <li>2. Click notification or open the app</li> <li>3. Should be presented with an incoming virus/vaccine</li> <li>4. Click “Accept” button</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>● User is playing and part of the same faction as the hot zone the user has entered.</li> <li>● User has received a new virus/vaccine</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>● A user becomes an active host of a strain.</li> </ul>

<i>Use Case Name</i>	Cure virus
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. Win fight minigame against a virus</li> <li>2. Receive cure</li> <li>3. Create cure</li> <li>4. Deploy Cure</li> </ol>

<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• The user must be a member of VacSect</li> <li>• The user must actively enter a hot zone or come into contact with a virus.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• The hot zone becomes a quarantine area that is reflected for all users.</li> </ul>

<i>Use Case Name</i>	Place BioBomb
<i>Flow of Events</i>	1. Place a Bio Bomb via the UI
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• A user is a member of BioHydra.</li> <li>• A user has a BioBomb in the inventory.</li> <li>• A user is walking around with location data turned on and enters a location where the bomb is deployable.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• A radial area of the deployment becomes a hot zone.</li> </ul>

<i>Use Case Name</i>	Place VacBomb
<i>Flow of Events</i>	1. Place a Vac Bomb via the UI
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is logged in and possesses a VacBomb in their inventory.</li> <li>• The User is a member of VacSect</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• Deployed area is turned into a large quarantine.</li> </ul>

<i>Use Case Name</i>	Create Virus
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. Chooses to create a new virus via the UI</li> <li>2. Give the virus a name and sets up the virus's minigame board via the UI</li> <li>3. Submit the completed vaccine to the server</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is logged in and meets the faction rank requirement</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• Virus has been created</li> <li>• The user possesses the new virus.</li> </ul>

<i>Use Case Name</i>	Create Vaccine
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. Choose to create a new vaccine via the UI</li> <li>2. Give the vaccine a name and sets up the vaccine's minigame board via the UI</li> </ol>

<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is logged in and meets the faction rank requirement</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• Vaccine has been created</li> <li>• The user possesses the new vaccine.</li> </ul>

<i>Use Case Name</i>	Use Consumable
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. View the inventory screen.</li> <li>2. Select an item to use.</li> <li>3. Select a way to deploy the consumable</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is logged in and has a consumable in the inventory</li> <li>• User is at a high enough level to use the consumable.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• A new strain has been created</li> <li>• The user possesses the new strain</li> <li>• The user no longer posses the consumable.</li> <li>• The user gains XP</li> </ul>

<i>Use Case Name</i>	Purchase Consumable
<i>Participating Actors</i>	Initiated by User Communicates with Server
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>9. Open the game menu and navigate to the Pandemic store.</li> <li>10. Find an item that the user can afford and click purchase.</li> <li>11. Confirm or Deny the purchase.</li> <li>12. Navigate to the inventory</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User has is a high enough level to purchase the desired consumable.</li> <li>• User has enough currency to afford the consumable.</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• Used currency is subtracted from the user’s wallet.</li> <li>• New consumable is added to the user inventory.</li> <li>• If the user selected “Deny,” do not charge the user and do not add the item to inventory</li> </ul>

<i>Use Case Name</i>	Rank Up
<i>Flow of Events</i>	<ol style="list-style-type: none"> <li>1. Perform an in-game task</li> <li>2. Confirm that the XP bar has been filled</li> </ol>
<i>Entry condition</i>	<ul style="list-style-type: none"> <li>• User is logged in and a faction member</li> </ul>



	<ul style="list-style-type: none"> <li>• User has gained enough XP to rank up</li> </ul>
<i>Exit conditions</i>	<ul style="list-style-type: none"> <li>• User rank increases by 1.</li> </ul>

## Testing schedule

- Automated tests on the AWS servers will run once per day to ensure any server updates have not broken any functionality and to ensure the server is not in an erroneous state.
- Human testers will test the product before each public release.

## Project Issues

### Open Issues

- Play regions do not currently have a determined size, and will need to be tested in order to find appropriate ranges.

### Off-the-Shelf Solutions

#### Ready-Made Products

- Google Maps
- Phone's built-in GPS system

#### Reusable Components

All of the ready-made products and libraries already mentioned in this report can be considered reusable components. As *Pandemic* is the company's first project, there is no existing company codebase to reuse.

#### Products That Can Be Copied

There are no comparable products which do everything that *Pandemic* requires. However, there are products we can use as examples of how to properly execute various aspects of *Pandemic*'s design. For example, Pokemon Go has been used regularly throughout *Pandemic* to illustrate certain aspects of the game.

## **New Problems**

### **Effects on the Current Environment**

There is no current environment, *Pandemic* is a stand-alone game that does not affect any existing systems.

### **Effects on the Installed Systems**

Pandemic should be entirely self-contained and not affect any other software or hardware.

### **Potential User Problems**

There are no existing users of *Pandemic*, it is a brand new game and is not replacing any existing system.

### **Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

Creating a globally-interactive game like Pandemic will provide challenges in deployment and testing.

### **Follow-Up Problems**

Pandemic *might* give real-world bioterrorists ideas, which among other things would have legal backlash. Try to avoid by making the game as clearly fictional as possible, as well as making a clear disclaimer. Creating a globally-interactive game like Pandemic will provide challenges in deployment and testing.

## **Tasks**

### **Project Planning**

Pandemic will use the standard Agile process during development, using the phases listed in the “Planning of the Development Phases” section below.

## Planning of the Development Phases

1. Hardware Appropriation: Obtain an iPhone for every iOS developer and an Android phone for every Android developer. Also obtain small quantities of legacy devices to test backwards compatibility. Purchase a small amount of Amazon AWS servers for development.
2. Server Development: Design schemas for all databases that will exist on the servers. Connect servers to all the APIs they will be using. Write all data insertion and extraction methods that will be called by mobile devices.
3. Mobile Development: Design the user interface and implement it for each mobile device. Write example data (that will come from the server in the future) and ensure the mobile devices properly display that data.
4. Integration Development: Ensure all kinds of mobile devices can successfully connect to the database. Test data insertion and extraction with mobile devices.
5. Scaling: Ensure that all aspects of the code will work under the stress of a massive user base. Procure more databases and ensure data is consistent between them.
6. Tester Deployment: Deploy the application to many servers and to the devices of beta testers.
7. Major Bug Fixes: Fix any tester-reported bugs as they come in.
8. Release: Release the product to the general public.

## Migration to the New Product

### Requirements for Migration to the New Product

Pandemic will require no migration, as it is exclusively a consumer product meant for entertainment.

### Data That Has to Be Modified or Translated for the New System

N/A

### Risks

- Cost estimation: if our group has not successfully estimated the cost of the entire project, this could result in a loss of the excess expenditure.
- The project may also have legal costs that could arise.

## Costs

The total estimated cost of the project is 4.20 million dollars, as mentioned in the Budget Constraints portion of the report.

## Waiting Room

- Add more items aside from BioBombs and VacBombs

## Ideas for Solutions

Server hosting will likely be solved using Amazon AWS, here's an introduction to it:

<https://aws.amazon.com/getting-started>

Google's Location based APIs will be used for many location services:

<https://developers.google.com/maps/>

## Project Retrospective

Some of the biggest mistakes our project designers made was getting too far ahead of ourselves. This includes things like designing implementation details before having a completely solid idea of how the game is actually to be played. However, we were able to finally settle on all of the gameplay details by the end of the project's design. This is a problem that I saw when other projects were presented to our group; By the end of their presentation, I didn't know how to play their game or what it was about but I did know how they were going to implement it. I think this problem is mostly a result of disorganization in the deadlines that the project's designers were required to meet.

## Glossary

**BioHydra:** One of the two opposing factions in *Pandemic*. Wants to spread their viruses to as many people as possible.

**VacSect:** BioHydra's opposition. Wants to spread their Vaccines to as many people as possible.

**BioBomb:** BioHydra item that, when placed, will infect users who pass by it.

**VacBomb:** The VacSect equivalent of a BioBomb.

