PROJECT DESCRIPTION

I. **Project Overview**: The idea is a kingdom conquering game using google maps API. The game is multiplayer where each player has a king/kingdom and soldiers. The kingdom provides defense for the city and must be protected. Soldiers are need to defend a kingdom or attack other kingdoms. The king orders soldiers to either attack, defend, standby, or retreat. In order to win the game, the surrounding areas must be conquered.

As google maps api is used, every town is a kingdom, and towns are able to attack surrounding kingdoms(towns). Also, using the weather and traffic from google maps API, it will provide obstacles for soldiers. The paths to other kingdoms will utilize major highways and major roads, not alleyways or residential areas.

2 The Purpose of the Project

SV: Describe WHY this project is being done, and what one hopes to achieve from it.

- To provide a challenging, yet entertaining game focused on building skills which you would need in any fast paced environment. The project is built around quick thinking and strategy.

2a The User Business or Background of the Project Effort

SV: Describe the client's business, e.g. the newspaper publishing business or the firefighting business, to the extent that it is relevant for this project. Note the distinction between "business" and "work" as described below in section I.3 below.

- **Content:** The users interest in games triggered this development project. The user intends to use this delivered product and mass sell it to valuable customers that are into competitive esports and more experienced players.
  - **Motivation:** To build a foundation by which we can expand upon the business and get attention of AAA gaming companies. This will allow us to expand and increase out PR. This will also help us in integrating augmented reality and virtual reality.
  - **Consideration:** Does not apply.

2b Goals of the Project

SV: Describe WHY this project is being carried out, from the point of view of the client. Note that the goal should be to improve the life of the client in some way, not just the development of software. (The SW is a means to an end, not the goal.) **12** ***(Note: This item and the following one together cover the " Objectives and success criteria of the project" item specified by Bruegge & DuToit.)***

- To look for creativity from smaller studios and find a new Intellectual Property for them to invest in. Hot market for these types of games. For example use: Pokemon Go. The product is wanted by the client to explore smaller creative studio ideas and appeal to the large mobile gaming market.

2c Measurement

SV: How will one know when the goals stated in I.2.2b have been met? What measurable result can we point to and say that the goal has been met?

- Measure the number of downloads per month and revenue generated from ads. Number of active players per month.

3 The Scope of the Work

SV: The "work" is a subset of the "business", and describes the set of activities that will be addressed by the proposed product. For example, if the business is "university-level education", then the work addressed by this project might be "the production and delivery of classroom lectures". Obviously the business of running a university encompasses a lot more than just classroom lectures, but this particular project will only concern itself with that particular aspect of the overall business.

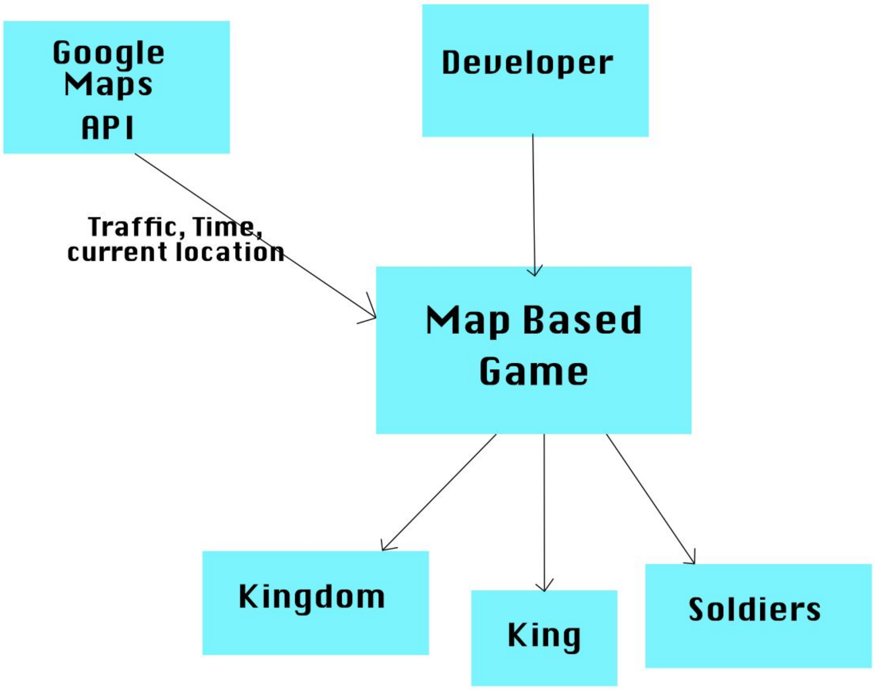
- The business is universal, but constrained by age (teenage and up) or any group. Meant for a competitive as well as casual audience that is fun to play and also challenging and fast paced.
-

3a The Current Situation SV: Describe how the client is conducting the work now, without the proposed product. Note that the current situation may or may not involve computers.

- Because this is a game, it doesn't change current business practices.

3b The Context of the Work ***SV: Define the boundary between what is included in "the work" and what is not. It also defines what external entities "the work" must interact with and what those interactions entail. The following example diagram should be replaced with one appropriate to this project.***

- There is the user that has a kingdom.
- Google maps api - traffic, time to get to destination, current location
- Developer
- The Game itself
    - Kingdom
    - King
    - Soldiers
- The UI



3c Work Partitioning

SV: "The work" is often large and complex, with many different activities and concerns. One good way to break this down and organize it for analysis is to identify the different events to which the business must respond. A "business event" is an external stimulus which causes the business to take a series of actions in response.

Event Name	**Input and Output**	**Summary**
1. Google API connects 2. Developer	- Read in traffic input - Displays paths - Display Current location - Requests google maps info	- Sends current time traffic report to update traffic levels - Outputs all possible pathways to destination - Shows current location of kingdom and soldiers - Displays map that includes major roads and highways

3d Competing Products ***SV: IF there are other existing products that the client could use instead of the proposed product, then they should be discussed here, along with the reasons why the proposed product is still needed / beneficial***.

There are other games out there such as Pokemon Go. However our game stands out as being competitive for players, appealing to a larger audience. Those who enjoy mobile games and those that enjoy intellectually stimulating games. Our game is more agile as it uses factors from Google API such as traffic, which allows our game to be constantly updated to the current data.

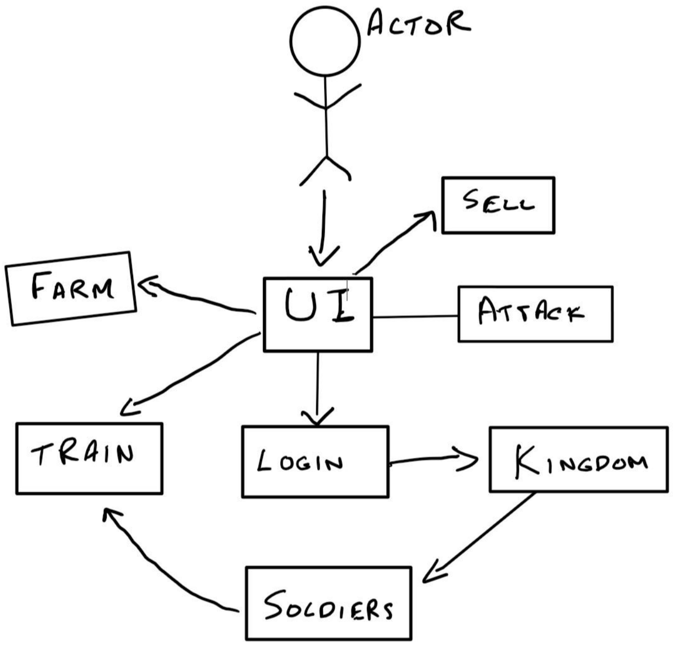
4 The Scope of the Project

The objective of the game is to conquer all the kingdoms that are on the current map. To begin, a user would have to first create an account through the UI. Once created, they will be given a small kingdom, along with a few soldiers. From there, the player can choose to do different tasks. They can either, train their soldiers, grow crops, or attack the other kingdoms. If a player chooses to train their soldiers, then that soldier cannot be used to attack or defend their kingdom. Once the training is complete, which will take 60 seconds, the soldier's attack and defense will increase by one. If the player chooses to grow crops, it will take 5 minutes for the plants to be ready for harvest which can then be sold for money. Each crop will sell for 5 gold coins. With the coins, the player can buy more soldiers. Lastly, when choosing to attack other kingdoms, the player will use all available soldiers to attack the other kingdom. If the player wins, then they have control of that land and all the soldiers that were previously under the enemy kingdom. If they lose, then they lose half their soldiers and must return to their kingdom.

4a Scenario Diagram

4b Product Scenario List

- User creation
- Building kingdom and giving soldiers
- Farming
- Training soldiers
- Selling/buying seeds and/or soldiers



- Attacking enemy kingdoms

4c Individual Product Scenario

- User creation: The user will create a user name and password. Once they do that, they will be given a small plot of land with few soldiers and some seeds to be used for farming.
- Farming: The user can use their seeds to farms. Once the plants are ready in 5 minutes, they can harvest them.
- Training soldiers: The soldiers can be trained to increase attack, defense, and HP. when training a soldier, they cannot be used to attack or defend their kingdom.
- Selling/buying seeds and/or soldiers: The market will be used to sell the crops or soldiers to get money. Seeds and soldiers can be purchased with the money.
- Attacking enemy kingdom: Player can choose which kingdom they want to attack, how many soldiers to take with them and how many soldiers to leave behind.

5 Stakeholders

5a The Client

SV: The client pays up front for the product to be developed, and provides guidance or other input for its development. Some projects do not have an external client, in which case the developing organization acts as the client.

-The client agreed to some terms after multiple negotiations. The client pays 100,000 to start developing this game. Then will pay 20% of the profits after the game has been developed and started selling.

5b The Customer

SV: The customer is the person or entity who will buy the product after it has been completed. Some projects do not have an external customer, if they are to be used in-house or for the client's use only.

-The customer are the people that will buy from the client. The client is used as a "middle man" to help sell this product with better advertising. The customers will know both the developing company and the client company.

5c Hands-On Users of the Product

SV: These are the people who will actually use the product in practice, and who may be separate from the customer or client. For example, educational software may be purchased by the school system (customer) and used by students (hands-on users.)

-There are no hands-on-Users of the product except for couple of people the client chooses for recreational purposes.

5d Maintenance Users and Service Technicians

SV: Describe users that will install, maintain, update, and otherwise service the product as needed. May not apply to all projects.

-The maintenance team will send out updates every week and the client and the customers just have to update the game from their local devices. The service team will be sent only there's a dire need of it.

5e Other Stakeholders

SV: This section is a catch-all for all other stakeholders not previously mentioned. Note that some stakeholders may be negatively impacted by the proposed project, for example if their work duties change or are eliminated. Some may even object to the project all together.

-None

5f User Participation

SV: To what extent can we expect users to participate during the development of the product?

-The users can occasionally make an appointment to check how the game is working within the headquarters of the development team. This will give them a chance to propose or advise to the hearing team about the game.

5g Priorities Assigned to Users

SV: To the extent that some users are more important to the project than others, the relative priorities should be identified here.

-The only priority are the users.

6 Mandated Constraints

6a Solution Constraints

SV: These are general constraints on the product to be developed or the manner in which it is to be developed that are not covered elsewhere.

The product can only be used on 4G LTE data connections or Wifi

6b Implementation Environment of the Current System

SV: This section deals with the physical and technical environment in which the proposed product will operate, such as hardware, operating system, and communications issues.

Only mobile devices with internet access will be able to access the information for the game, and play the game.

6c Partner or Collaborative Applications

SV: This section documents external applications with which this product must be compatible, such as the ability to read and write Microsoft Excel format data files.

-This project will interface and collaborate with Google Maps.
-This project will interface with The Weather Channel's live data.

6d Off-the-Shelf Software

SV: This section describes commercial off-the-shelf (COTS) software that MUST be included in the final product.

There will be no extra needed software beyond what is developed specifically for this project.

6e Anticipated Workplace Environment

SV: This section deals with human factors regarding the environment in which the product will be used, such as noisy environments or mobile applications.

There are limitations on the locations of the events and marked locations for this product. But directions will be limited to major roads and highways.

6f Schedule Constraints

SV: When things must be done, or when they may be most/least beneficial.

There are no scheduling constraints as the product will be available for use 24/7.

6g Budget Constraints

SV: Limitations on the funds and other resources available for this project.

The budget for this product is a total of $1.5 million.

7 Naming Conventions and Definitions

SV: Define terminology to avoid miscommunications or misunderstandings.

7a Definitions of Key Terms

SV: Define words that may have special or multiple meanings.

HP - Health points for each soldier

STR - Soldier level of strength. Decides the amount of attack

AGL - Soldier Speed

DEF - Defense amount for kingdom and soldiers

Path - the path from one kingdom to another

Danger Level - the danger level on the path

7b UML and Other Notation Used in This Document

Following the UML Diagram notation from Martin Fowler's "UML Distilled: Third Edition". Symbols such as aggregation, composition, inheritance, interface etc. will be utilized from this source.

7c Data Dictionary for Any Included Models

SV: Define ***data structures*** *and* ***data properties*** *relative to this project, such as the contents of an employee record or the fact that student GPA ranges from 0.0 to 4.0 corresponding to letter grades of F to A. Data file formats may be referenced to documented standards, such as jpg or pdf.*

Danger Level = the traffic level from Google Maps API (High traffic = "more danger") + Time (Certain times of day are more dangerous than others i.e. night time more dangerous than daytime) + Weather in Area

Will define more details later on as Project Description is defined more in depth…

8 Relevant Facts and Assumptions

8a Facts

SV: Factual information relevant to the project, such as census data.

Google Maps data will need to be utilized for paths and danger level. Your own town will be used as one kingdom, while the surrounding towns will be kingdoms for others players.

8b Assumptions

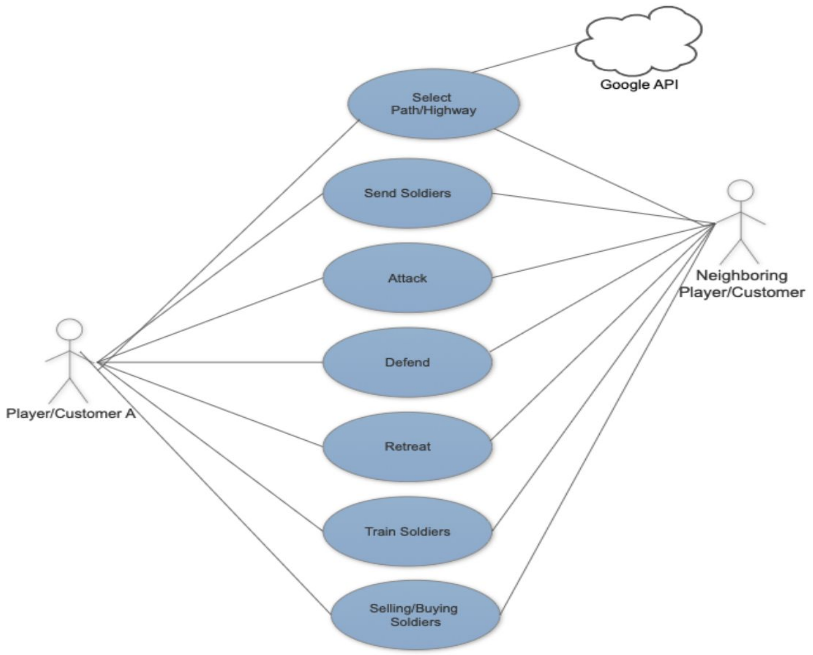
SV: Assumptions relevant to the project, such as the availability of necessary resources or abilities of the users.

- Assuming the user is in a fairly populated area with paths and traffic
- Assuming the system can only be used on 4G LTE data connections or Wifi
- Assume the player is teenage or older

II Requirements

9 Product Use Cases

9a Use Case Diagrams



USE CASE DIAGRAM: Clash of Kingdoms

The use case diagram shows the main external actors which are the players. The players interact with a variety of commands. Also, the Google API is utilized for data such as highway paths, and constant traffic (or danger level) updates.

9b Product Use Case List

Not applicable because there are currently 7 use cases. In a case where there are 15+ use cases a use case list would be more applicable.

9c Individual Product Use Cases

Use case ID: 1 Name: Selecting Highway/Path
pre-conditions: Paths are created from Google API data
post-conditions: User is prompted to pick which soldiers to send
Initiated by: Player
Triggering Event: N/A
Additional Actors: N/A

Use case ID: 2 Name: Send Soldiers
pre-conditions: Player must either by looking for paths, attacking, or own multiple kingdoms
post-conditions: Soldiers are sent from players kingdom to an opposing kingdom/to another kingdom they own
Initiated by: Player
Triggering Event: N/A
Additional Actors: N/A

Use case ID: 3 Name: Attack
pre-conditions: Player has soldiers available
post-conditions: Player's soldier count inside the kingdom decreases
Initiated by: Player
Triggering Event: N/A
Additional Actors: N/A

Use case ID: 4 Name: Defend
pre-conditions: Player has soldiers available
post-conditions: Player's soldier count inside the kingdom decreases
Initiated by: Player

Triggering Event: Enemy spotted, approaching, or attacking on of the player's kingdoms
Additional Actors: N/A

Use case ID: 5 Name: Retreat
pre-conditions: N/A
Post-conditions: Soldiers return to the kingdom
Initiated by: Player
Triggering Event: Loss of soldiers or the kingdom is in danger or reallocation of resources
Additional Actors: N/A

Use case ID: 6 Name: Train
pre-conditions: Player has soldiers available
post-conditions: Soldier's available to send out decreases
Initiated by: Player
Triggering Event: N/A
Additional Actors: N/A

Use case ID: 7 Name: Selling/Buying
pre-conditions: N/A
post-conditions: Player loses or gains money
Initiated by: Player
Triggering Event: Need extra soldiers and crops or money
Additional Actors: N/A

10 Functional Requirements

ID# 10a - Choose Area

Description: The program must allow player to choose their kingdom and the area of the kingdom.

Rationale: The player must be able to choose where they want to start, but it must be their local town. This will affect which paths the player can take.

Fit Criterion: The requirement will be met when the Google API loads the map.

Acceptance Tests: Check section 21

ID# 10b - Danger

Description: The system must allow for the ability to tell the user/king if there's danger approaching

Rationale: This will allow users to change their strategy.

Fit Criterion: The product requirement will be met when the product fully detects oncoming soldiers and updates there movements correctly.

Acceptance Tests: Check section 21

ID# 10c - Weather

Description: The system must update weather every couple of minutes

Rationale: Player needs to see changes in environment to change their strategy

Fit Criterion: The product requirement will be met when the system shows the weather every couple minutes, and the report is completely accurate.

Acceptance Tests: Check section 21

ID# 10d - Map

Description: The program must display a map with neighboring highways and towns

Rationale: Player needs to see the different path options and kingdoms

Fit Criterion: The product requirement will be met when the a map is displayed with correct highways and towns/kingdoms are distinct.

Acceptance Tests: Check section 21

ID# 10e- Track Danger

Description: The program must keep in track of the traffic (danger) levels on highways and update every 30 minutes or so.

Rationale: Player needs to see the danger level to predict future strategies

Fit Criterion: The product requirement will be met when the danger level is calculated using the weather and traffic levels, and is consistent, going up when it’s harsher weather and higher traffic, every 30 minutes for a week or two

Acceptance Tests:

ID# 10f - Notify Deceased

Description: The system must notify the user when soldiers die

Rationale: This will either end the game for the player or allow them to change their strategy

Fit Criterion: The product requirement will be met when a message pops up each time a soldier dies and the army number decreases. This will be tested for instantaneous response as to notify the player as soon as it happens.

Acceptance Tests: Check section 21

ID# 10g- Notify Victory

Description: The system must notify when a neighboring player is conquered

Rationale: The player then takes control of the neighboring kingdom and gains the soldiers and money. It's closer to winning the game.

Fit Criterion: The product requirement will be met when a message pops up each time an enemies kingdom is conquered. This must be instantaneous and will be tested for correctness.

Acceptance Tests: Check section 21

ID# 10f - Active Soldiers

Description: The system must check the number of soldiers the currently in training and on the battlefield

Rationale: The player must know which soldiers can be utilized.

Fit Criterion: The product requirement will be met when a players are not allowed to deploy training soldiers

Acceptance Tests: Check section 21

11 Data Requirements

ID# 11a - Save Info

Description: The system must record each unique username and password
Rationale: The product must save usernames and passwords for logging in again
Fit Criterion: This system will be tested by adding and retrieving thousands of random usernames and passwords
Acceptance Tests: Check section 21

ID# 11b - Unique Info

Description: Database will be checked for repeated usernames, each new username must be unique
Rationale: The product must not have any repeat any usernames
Fit Criterion: This system will be tested by adding and retrieving thousands of random usernames and passwords. Each time a username matches in the database a pop up should check for repeats.
Acceptance Tests: Check section 21

ID# 11c - Google API

Description: Google API must provide traffic, town, and highway information

Rationale: The information is important for building the UI

Fit Criterion: N/A

Acceptance Tests: Check section 21

ID# 11d - Weather API

Description: Weather API must keep track of weather data: rainy, stormy, snowing, etc.

Rationale: The information is important for calculating danger levels

Fit Criterion: Weather data will be monitored and checked for accuracy every 30 minutes using local readings for a couple weeks. Once readings show an accurate match then the weather API can be utilized.

Acceptance Tests: Check section 21

12 Performance Requirements

12a Speed and Latency Requirements

ID#12a - Speed

Description: The System may take less than 5 seconds to configure the location and time, but should not take more than 1 second to do basic functionality.

Rationale: The delay is done to make sure that the user has a smooth transition from one area to the other.

Fit Criterion: The system must be taking less than 1 second 90% of the time. The exception of going over 1 second should be only for updating while the user is not using their phone.

Acceptance Tests: Compare locations with google API.

12b Precision or Accuracy Requirements

ID#12b - Precision

Description: All units will be in the metric system. And the destination should be just meters away from the actual destination.

Rationale: The descriptions is ensured with better algorithms and using the google satellites.

Fit Criterion: The distance from one place to another should have time +-2 minutes. And the destination should be around 2 meters from the actual destination.

Acceptance Tests: The product might also need to keep accurate time, be synchronized with a time server, or work in UTC.

12c Capacity Requirements

ID#12c - Capacity

Description: The product has at most 50 players playing within a multiplayer game.

Rationale: As the game changes, your surrounding improves and you are put into a bigger group with harder challenges. But the system only allows for 49 other challengers at once.

Fit Criterion: The description was quantified.

Acceptance Tests: Try adding more players than the max. And it shouldn't work.

13 Dependability Requirements

13a Reliability Requirements

ID#13a - Reliability

Description: The product should not force exit more than once a day if needed.

Rationale: Force exit allows the system to lose it data for the few instances before it was forced to exit. The data otherwise is safe and never deletes itself.

Fit Criterion: Every 24 hours, apart from saving within the device, the data should be sent up to cloud for more protection of it.

Acceptance Tests: The product should be kept on for 24 hours and keep using to see any forced exit.

13b Availability Requirements

ID#13b - Availability

Description: The product shall be available for use 24 hours per day, 365 days per year.

Rationale: The product should never have downtime, unless the system is updating which was written off by the user.

Fit Criterion: The description was quantified.

Acceptance Tests: Keep the product on for 24 hours for 7 days.

13c Robustness or Fault-Tolerance Requirements

ID#13c - Robustness

Description: If the system is offline, for example does not have an internet connection, it should not make any changes within the product.

Rationale: The user shouldn't be able to proceed with more milestones without internet. They can see their previous achievements, but cannot proceed.

Fit Criterion: The product tries to connect to hotspot within the range of .25 miles if the internet from the user is not detected or unable to connect.

Acceptance Tests: Turn the Wifi on and off. Then try to connect with 50 hotspots throughout each city.

13d Safety-Critical Requirements

ID#13d - Safety-Critical

Description: The system may have been designed in a way that uses the user's outside environment, but it should not encourage the user to not pay attention on their surrounding.

Rationale: The product may be fun, but lives are important.

Fit Criterion: The systems checks every 30 minutes if the game is being played while driving. Reminds the user every time to set their phones aside.

Acceptance Tests: Drive while using the product.

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

ID#14a - Maintenance

Description: The system tries to update any api that needs to be updated overnight.

Rationale: This ensure for a complete convenient experience with the product as it's being used throughout the day,.

Fit Criterion: The actual system updates every 1 month and the users are notified of it beforehand.

Acceptance Tests: Try to update the product for 30 days straight with minimal changes. Set the time.

14b Supportability Requirements

ID#14b - Supportability

Description: The product allows for support through a connect button within that allows the user to send an email to the team that takes the complaints or suggestions.

Rationale: There will also be a phone number that the users can call to ensure customer service is adequately provided.

Fit Criterion: The suggestions are read every morning Mon-Fri. The complaints are dealt with on the day of if possible, otherwise a new case is opened for the engineers.

Acceptance Test: Email with private and public accounts.

14c Adaptability Requirements

ID#14c - Adaptability

Description: The system was designed to work with windows XP, mac, linux.

Rationale: This is required to ensure no one is left out of this experience. There are three different teams that ensure adaptability within the user's systems

Fit Criterion: The product is offered within all software systems up until 2018.

Acceptance Tests: Try using it in different software and systems.

14d Scalability or Extensibility Requirements

ID#14d - Scalability

Description: The system shall be improved to take more than 50 players within one game.

Rationale: Scalability allows for more diverse players going against each other.

Fit Criterion: This number of players should double within the two years of its release.

Acceptance Tests: Update game and try to add 1 player at a time while the game is running.

14e Longevity Requirements

ID#14e - Longevity

Description: This product will be self funded. But without the extra funds created, the product is well funded for the next two years.

Rationale: Money is essential with the improvement, supportability and maintenance. And the team of financials were also looked into to maintain a book of costs to expense.

Fit Criterion: The description was quantifiable.

Acceptance Tests: None.

15 Security Requirements

15a Access Requirements

ID#15a - Access

Description: This product will allow users to only have access to their own account information such as password and email address.

Rationale: Getting access to other user's information such as email address and password can give allow others to manipulate other users' account.

Fit Criterion: When a user clicks on another user, they should not be able to see any other information besides username and other game related information.

Acceptance Tests: Hire a team of hackers to try and exploit the product and retrieve player information such as password and email address. This will occur every couple months.

15b Integrity Requirements

ID#15b - Integrity

Description: The product will not allow multiple usernames that are the exactly the same.

Rationale: Having multiple same usernames will corrupt the data.

Fit Criterion: Before an account is created, the username will be checked against the database to see if that username already exists.

Acceptance Tests: Hire a team of hackers to try and exploit the product and retrieve player information such as password and email address. This will occur every couple months.

15c Privacy Requirements

ID#15c - Privacy

Description: This product shall let users know that their email address will be stored and in case of forgetting their password, the product will send out an email to the user with a temporary password.

Rationale: The user must be aware of whenever their information is being used.

Fit Criterion: When creating the account, the users will be required to agree on the terms in order to create their account.

Acceptance Tests: Hire a team of hackers to try and exploit the product and retrieve player information such as password and email address. This will occur every couple months.

15d Audit Requirements - NOT REQUIRED

ID#15d - Audit

Description: None**.**

Rationale: None**.**

Fit Criterion: None.

Acceptance Tests: None.

15e Immunity Requirements

ID#15e - Immunity

Description: This product will use cookies to prevent malicious activity.

Rationale: Cookies will give each user a unique code whenever they login. That way they can be the only ones accessing their account at that particular moment.

Fit Criterion: Each user will be given a cookie as well as have their IP address stored. This will be compared every time they login. If the IP address does not match then an email will be sent for confirmation.

Acceptance Tests: Hire a team of hackers to try and exploit the product and retrieve player information such as password and email address. This will occur every couple months.

16 Usability and Humanity Requirements

16a Ease of Use Requirements

ID#16a - Easiness

Description: This product shall be easy to use once the user completes a basic tutorial.

Rationale: The tutorial will go over all the rules and once the tutorial is completed the game will be easy to play and understand.

Fit Criterion: Once the tutorial is completed, the user will not have to use the tutorial once again. An anonymous survey will tell whether or not a user used the tutorial more than once.

Acceptance Tests: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.

16b Personalization and Internationalization Requirements

ID#16b - Personalization

Description: The product shall allow the users to select whether they want to play the game in English or Spanish.

Rationale: This will allow the product to be used by a larger audience.

Fit Criterion: By getting IP addresses, we will be able to see where the users are located from and whether or not having two languages increased audience numbers.

Acceptance Tests: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.

16c Learning Requirements

ID#16c - Learning

Description: The product will be used by members of the public who will receive training from the tutorial that is part of the product.

Rationale: The tutorial will go over all the functionalities of the product.

Fit Criterion: Once the tutorial is complete, the user would not have to reference it ever again.

Acceptance Tests: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.

16d Understandability and Politeness Requirements

ID#16d - Understandability

Description: The product shall use symbols and words that are naturally understandable by the user.

Rationale: This will make the product more useable and easier to understand.

Fit Criterion: A survey will have images and words in the survey. 99 percent of the survey takers should correctly understand the meaning of the images and words.

Acceptance Tests: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.

16e Accessibility Requirements

ID#16e - Accessibility

Description: The product shall be used by people with bad vision.

Rationale: The users can adjust the text size which will allow them to read the text with more ease.

Fit Criterion: Users will be given a survey that asks if the text adjustment is beneficial.

Acceptance Tests: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.

16f User Documentation Requirements

ID#16f - Documentation

Description: The product shall provide users with an installation document.

Rationale: This document will make it easy for any user to install the program and and answer any frequently asked questions.

Fit Criterion: A survey will be used to see if the installation instructions made it easy or not for the user to use the product. One hundred percent of the results should say that it made it easy.

Acceptance Tests: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.

16g Training Requirements

ID#16g - Training

Description: The product shall provide a tutorial.

Rationale: The only training that users will receive will be the tutorial that the product provides.

Fit Criterion: The tutorial will make it easy for the user to use the product.

Acceptance Tests: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.

17 Look and Feel Requirements

17a Appearance Requirements

ID#17a - Appearance

Description: This product shall be attractive to a teenage and adult audience

Rationale: The goal is for competitive play so an older audience would be targeted

Fit Criterion: A sampling representative of teens and adults will be shown the game and will be timed for how long they play. More than 5 minutes is a success.

Acceptance Tests: Check section 21

17b Style Requirements

ID#17b - Style

Description: The product will feel cool and easy to use

Rationale: To appeal to a larger audience and compete with AAA games

Fit Criterion: To engage in customers and ask them to fill out a survey on the quality of the product in terms of graphics, sound, feel and ease of use.

Acceptance Tests: Check section 21

18 Operational and Environmental Requirements

18a Expected Physical Environment

ID#18a - Environment

Description: The game can work in any physical environment, indoor or outdoor

Rationale: The game uses weather API but will not be restricted by any conditions in nature

Fit Criterion: The description describes itself

Acceptance Tests: Any outdoor environment will suffice

18b Requirements for Interfacing with Adjacent Systems

ID#18b - Interface

Description: The game will work on the last few releases of the most popular operating systems for IOS and Android

Rationale: This is a mobile game and should always be up to date with the new mobile capabilities

Fit Criterion: Specific release version for mobile operating systems

Acceptance Tests: The OS release matches one on a short list of the most recent releases

18c Productization Requirements

ID#18c - Production

Description: Any user with access to an online app store can download the game

Rationale: The game is for the general public and should be simple

Fit Criterion: Self-explanatory

Acceptance Tests: Device has an app store

18d Release Requirements

ID#18d - Release

Description: Every new release should not affect any previous release

Rationale: The new releases should always remain compatible with existing features

Fit Criterion: Compare the new release to the previous ones for overlap

Acceptance Tests: No new bugs from what used to work

19 Cultural and Political Requirements

19a Cultural Requirements

ID#19a - Cultural

Description: All software shall be written solely in the company's development teams

Rationale: Keeps all the software rights to the company itself

Fit Criterion: None

Acceptance Tests: The code for the software should be tracked inside of the company

19b Political Requirements

ID#19b - Political

Description: Game should keep politically unbiased as possible

Rationale: Self-Explanatory

Fit Criterion: None

Acceptance Tests: Any mention of real political parties or ideas will not pass

20 Legal Requirements

20a Compliance Requirements

ID#20a - Compliance

Description: The game may not use anything from other software without rights to it

Rationale: To avoid lawsuits

Fit Criterion: Unique software from our own development team

Acceptance Tests: Code or ideas for software must be unique

20b Standards Requirements

ID#20b - Standards

Description: There should be some new features and usability at least every three months

Rationale: Change and new things keep the users inclined to look forward to new features

Fit Criterion: Every three months, a new release at the minimum

Acceptance Tests: Fails if there is not a new release every 3 months

21 Requirements Acceptance Tests

21a Requirements – Test Correspondence Summary

Test	Requirements																			
	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9	Req 10	Req 11	Req 12	Req 13	Req 14	Req 15	Req 16	Req 17	Req 18	Req 19	Req 20
Test 1	X																			
Test 2		X																		
Test 3			X																	
Test 4				X																
Test 5					X															
Test 6						X														
Test 7							X													

Test 8								X												
Test 9									X											
Test 10										X										
Test 11											X									
Test 12												X								
Test 13													X							
Test 14														X						
Test 15															X					
Test 16																X				
Test 17																	X			
Test 18																		X		
Test 19																			X	
Test 20																				X

21b Acceptance Test Descriptions

ID# - Name

Description:

10a: The test will check if Google API correctly loads the highways.

10b: All changes to the map must be shown in seconds

10c: When the current weather matches the weather API every 30 minutes for a couple weeks.

10d: Same as 10a.

10e: Same as 10a and 10c

10f: Test the notification system and see if correct message pops up when a soldier dies.

11a: This system will be tested by adding and retrieving thousands of random usernames and passwords

11b: This system will be tested by adding and retrieving thousands of random usernames and passwords. Each time a username matches in the database a pop up should warn about repeats.

11d: Weather data will be monitored and checked for accuracy every 30 minutes using local readings for a couple weeks. Once readings show an accurate match then the weather API can be utilized.

12a: Compare locations with google API.

12b: The product might also need to keep accurate time, be synchronized with a time server, or work in UTC.

12c: Try adding more players than the max. And it shouldn't work.

13a: The product should be kept on for 24 hours and keep using to see any forced exit.

13b: Keep the product on for 24 hours for 7 days.

13c: Turn the Wifi on and off. Then try to connect with 50 hotspots throughout each city.

13d: Drive while using the product.

14a: Try to update the product for 30 days straight with minimal changes. Set the time.

14b: Email with private and public accounts.

14c: Try using it in different software and systems.
14d: Update game and try to add 1 player at a time while the game is running.
14e: The description was quantifiable.
15a, b, c, d, e: Hire a team of hackers to try and exploit the product and retrieve player information such as password and email address. This will occur every couple months.
16a, b, c, d , f, g: Test the product in different countries and have them fill out a survey that rates the quality of understandability, accessibility, and overall ease of use on a 10 point scale.
17a: A sampling representative of teens and adults will be shown the game and will be timed for how long they play. More than 5 minutes is a success.
17b: To engage in customers and ask them to fill out a survey on the quality of the product in terms of graphics, sound, feel and ease of use.
18a: Any outdoor environment will suffice
18b: The OS release matches one on a short list of the most recent releases
18c: No new bugs from what used to work
19a: The code for the software should be tracked inside of the company
19b: Any mention of real political parties or ideas will not pass
20a: Code or ideas for software must be unique
20b: if there is not a new release every 3 months