Software Requirements Specification

for

Unconquered

**Prepared by:**

***For Lack Of a Better Name***

Matt Thompson

Matt Mariutto

Calvin Labbe

Michael Edgar

CEN4020 Software Engineering

Dec 5, 2014

1. **Introduction**
   1. **Purpose**

The purpose of this SRS is to provide a detailed and organized description of our application, “Unconquered”. It includes diagrams detailing the flow of data and how the different parts interact.

This document is intended for both business and technical facets of the application. While technical orientated readers may find some information redundant, it is important to remember that things may be explained again in a different context for a different audience. We hope that this document will be an informative and useful guide for those maintaining “Unconquered”.

* 1. **Scope of the System Specified**

The purpose of “Unconquered” is to create a fun and adventurous world on a mobile platform for players to immerse themselves in by seeing more of the world around them. As players perform real world actions and record them using their Android device, they are rewarded with increases in character attributes such as strength, intelligence, and charisma. When in close proximity to another user, a battle will initiate and the winner will be determined by character stats and probability. In order to win more battles, players will have to play frequently and level up their character.

* 1. **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| SRS | Software Requirement Specifications |
| UML | Unified Modeling Language |
| SQL | Structured Query Language |
| SDK | Software Development Kit |
| API | Application Program Interface |
| GPS | Global Positioning System |
| RPG | Role Playing Game |
| STR | Strength |
| END | Endurance |
| CHR | Charisma |
| INT | Intelligence |
| AGL | Agility |
| LCK | Luck |
| FSU | Florida State University |
| EXP / XP | Experience Points |

* 1. **References to Supporting Documents**

No other documents are referenced in this SRS. However, we did reference the slides from Dr. Gaitros’ software engineering lectures in order to learn how to prepare the UML diagrams, and this document.

Due to the nature of this program, we used several online sources for information about the Android SDK, Java, the Google Maps API, and SQL database.

* 1. **Overview of rest of SRS**

Everything that reveals essential information about what the application does and how is explained earlier. From here, this document gets more technical.

In the coming sections, one can expect to find information about everything the system needs to run correctly. After that, the document delves into the internal workings of the application, such as standard class diagrams, sequence diagrams, and other information relevant to how the system operates on a step-by-step level.

1. **General Description**
   1. **Product Perspective**

All users will begin the game by creating a username and password. The username must be unique. Once the user is signed in, the software will provide the user with their own information, such as skills, abilities, etc. If the user completes any of the predetermined actions, their skills will be incremented, depending on the action and how long or how frequently the user performs it.

When the app detects another user nearby, a battle will ensue. During a battle, player-queued actions will be executed and the outcome will be determined by skills and how they affect each other. The element of randomness in a battle is determined by the “Luck” stat, which will only have a minimal effect.

Outside of battles, users can set or change their queued actions, view their stats, chat with other users, and check other data.

* 1. **Product Functions**

The basic functionality of the game depends on user-activated functions: “WatchBattle”, “CheckIn”, “ViewStats”, and “ViewMap”. These functions alternate between screens of the application as well as alter the user data. “WatchBattle” will view previous battle sequences with other players in a nearby location. “CheckIn” will access data location and verify the location data with predetermined location values. This function will also alter the user’s character data in the database. “ViewMap” will just show the user a map of the local areas including any predetermined locations nearby.

* 1. **User Characteristics**

The intended users of our software are college students at FSU, and would expand to other locations as we expand the map and functionality of the game. The only technical expertise requirement is the ability to work an Android smartphone. Users will be localized to the Tallahassee area around Florida State University to begin with. Our game design will be based around our goal to entertain and maintain the intended users.

* 1. **General Constraints**

One concern in the software is the validity and security of the location data we gather from the user’s smartphone. We must be sure not to overstep any ethical or legal issues pertaining to users’ privacy. We would require users to agree to give the application access to certain functionality on their phone to accurately give their location to interact with the game interface, as well as allowing their device to communicate directly with other devices.

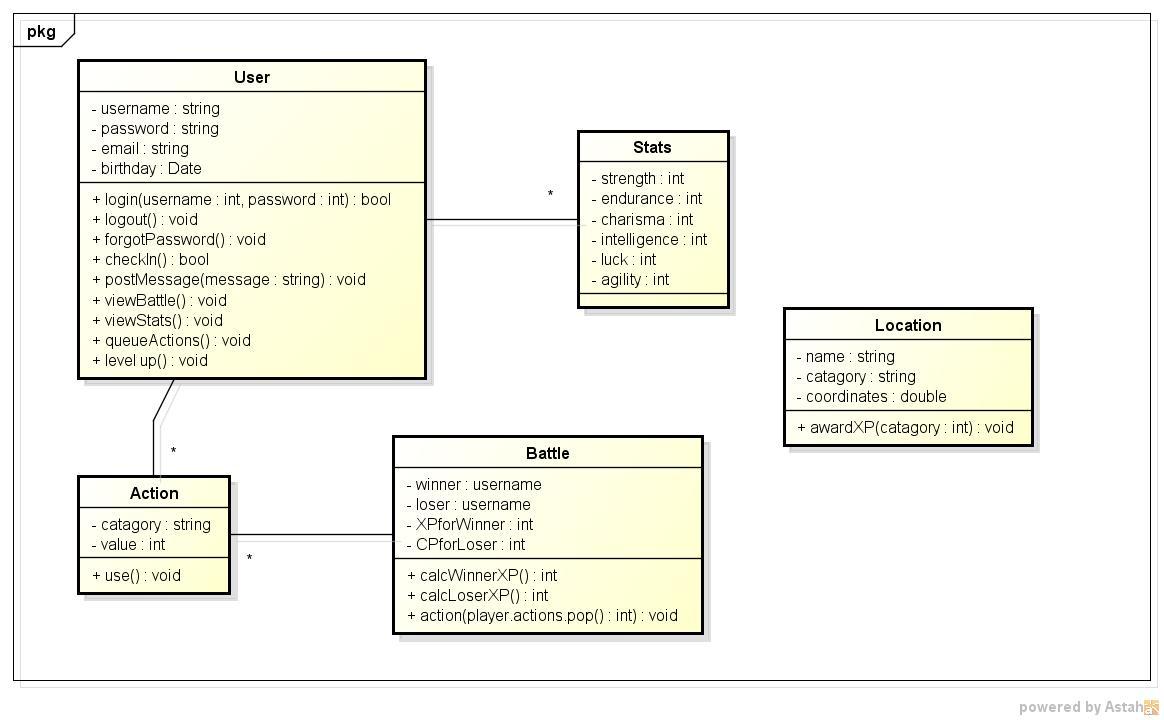
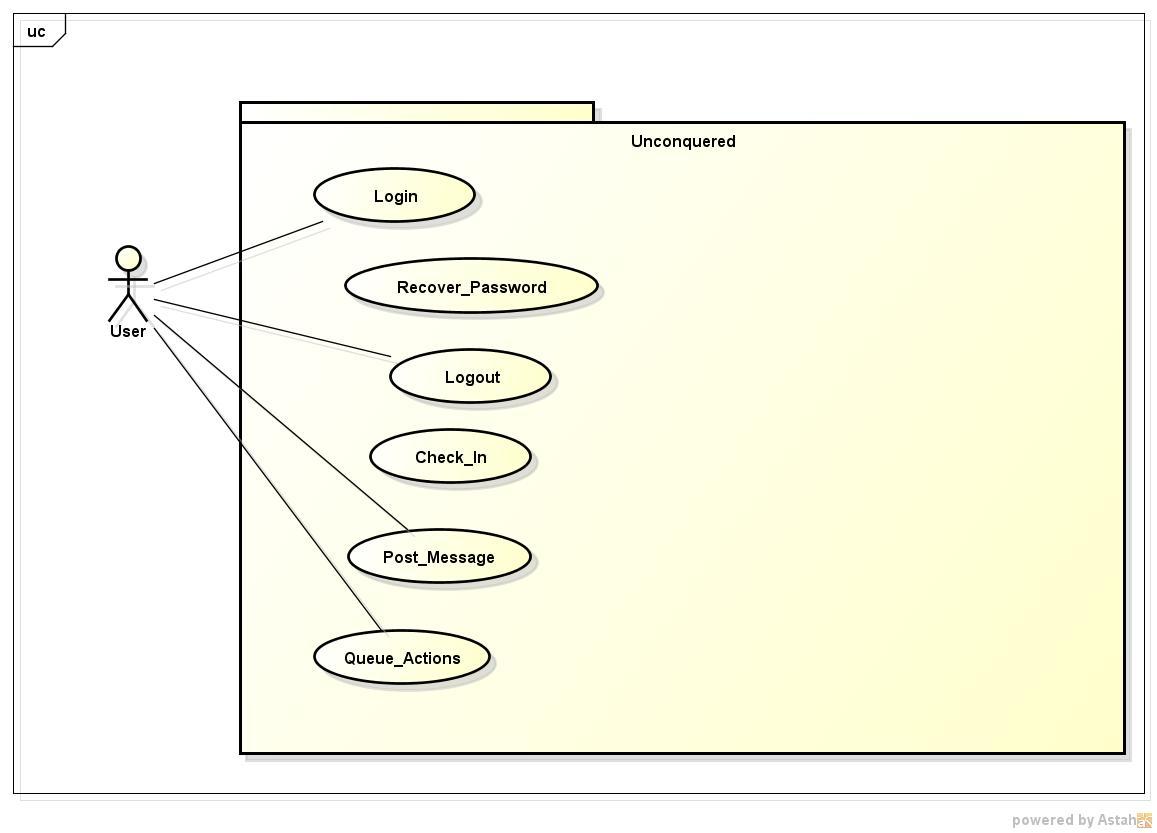
* 1. **Assumptions and Dependencies**

Assumptions:

* The device has cellular service or Wi-Fi Access to access mobile data.
* The device has access to a GPS.
* The device does not have any major breaks or hacks that would change the driver access.
* The device is being used by the legal registered owner of the device.
* The user can toggle on/off Bluetooth activity on their device.
* The user grants the application permission to access the necessary functions.

Dependencies:

* The Program requires the Android operating system.
* The Program requires mobile data for location.
* The Program requires Bluetooth functionality to locate other devices.
* The Program requires enough free memory to run.

1. **Functional Requirements**
2. The system shall have a main game, executing a series of predetermined inputs in battle.
3. Battles shall be initiated automatically when two users pass each other.
4. The system shall record battles when they happen.
5. The system shall playback the recorded battles for the player when requested.
6. The system shall allow the user to change the inputs for the battle based on the abilities they have available.
7. The system shall have statistics that are determined by the actions of the user. These include interacting on a geographic check-in system, a real-time chat, and other actions in the game.
8. The system shall have a geographic check-in system, allowing the player to register their location. Examples include gym, school, library, park, and others.
9. The system shall display a map to the user of the immediate area around the device’s current location.
10. The system shall have a real-time chat, displaying messages of all users in a localized area.
11. The system shall save the progress and actions of the user, in order to access the information in the form of statistics.
12. The system shall provide the users’ personal skills and available inputs to the user.
13. The system shall determine which inputs and abilities a user has, according to their level and skills.
14. **Non-functional Requirements**
15. All location data obtained from the device GPS must be accurate.
16. Data must be saved in a SQL database.
17. User data and connection must be encrypted and secure.
18. A user’s device must have mobile data.
19. A user’s device must have Bluetooth or wireless activated to connect to other devices.
20. **System Architecture**
21. **System Model**
22. **Appendices**
    1. **Data dictionary**
       1. **Actor Descriptions**

* User- can perform all of the functions necessary to play the game
  + 1. **Use Case Descriptions**
* Login- allows the user to sign in with their username and password
* Recover Password- allows a user who has forgotten their password to change it after verifying the account
* Logout- ends the session
* Check-in- the user checks-in to a location to gain XP
* Post Message- the user posts a message to the game’s message board
* Queue Actions- the user queues actions on a stack to be used in the next battle
  + 1. **Class Descriptions**
* User- abstraction of a user that is signed in and playing
* Battle- Event that occurs when two users are in close proximity
* Location- a point on the map the user can visit to earn XP
* Stats- attributes that define a player’s overall strength
* Action- items that can be queued for use in battle
  + 1. **Attribute Descriptions**
* username- a screen name the user makes that must be unique
* password- password to login
* email- user’s email address that is required to create an account
* birthday- user’s date of birth
* category- defines the category of the action (physical, magic, etc…)
* value- measures the strength of an action
* winner- user that won the battle
* loser- user that lost the battle
* XPforWinner- amount of XP that is awarded to the winner of the battle
* XPforLoser- amount of XP that is awarded to the loser of the battle
* name- name of the location (Library, Gym, etc…)
* category- stat that the location increases affects
* coordinates- coordinates of a location
* strength- measures the user’s strength
* charisma- measures the user’s charisma
* endurance- measures the user’s endurance
* intelligence- measures the user’s intelligence
* agility- measures the user’s agility
* luck- measures the user’s luck