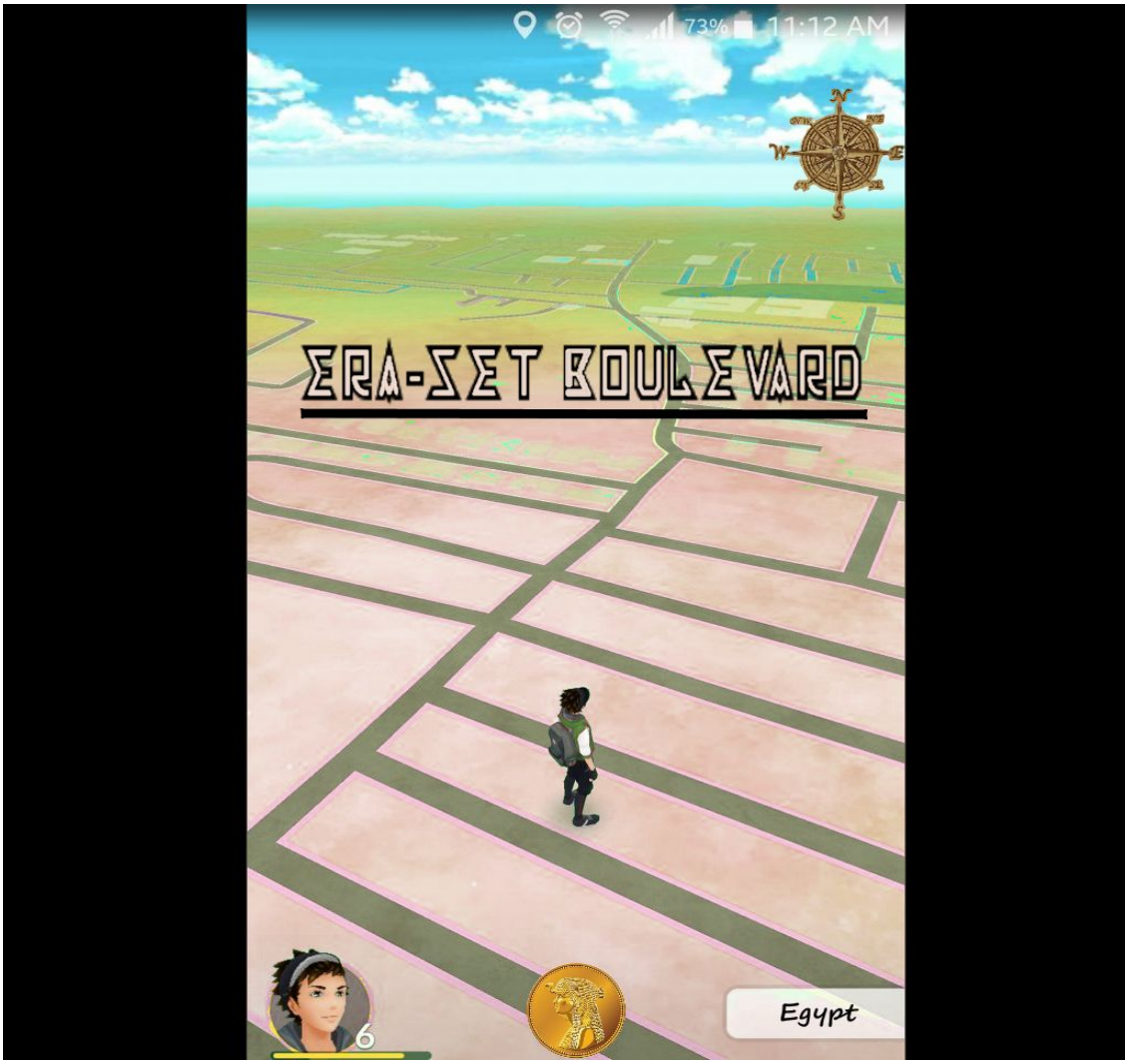
Software Engineering Project Report:
Era-set Boulevard



Prepared by
Constantino Rodriguez, Talha Taj, Nathan Tisdale-Dollah, Jon Tran
CS440
University of Illinois at Chicago
Fall Semester 2016

Table of Contents

7.b. UML and Other Notation Used in This Document
7.c. Data Dictionary for Any Included Models

8. Relevant Facts and Assumptions

8.a. Facts
8.b. Assumptions

II. Requirements

1. Product Use Cases

1.a. Use Case Diagrams
1.b. Product Use Case List
1.c. Individual Product Use Cases

2. Functional Requirements

3. Data Requirements

4. Performance Requirements

4.a. Speed and Latency Requirements
4.b. Precision or Accuracy Requirements
4.c. Capacity Requirements

5. Dependability Requirements

5.a. Reliability Requirements
5.b. Availability Requirements
5.c. Robustness or Fault-Tolerance Requirements
5.d. Safety-Critical Requirements

6. Maintainability and Supportability Requirements

6.a. Maintenance Requirements
6.b. Supportability Requirements
6.c. Adaptability Requirements
6.d. Scalability or Extensibility Requirements
6.e. Longevity Requirements

7. Security Requirements

7.a. Access Requirements
7.b. Integrity Requirements
7.c. Privacy Requirements
7.d. Audit Requirements
7.e. Immunity Requirements

8. Usability and Humanity Requirements

8.a. Ease of Use Requirements

8.b. Personalization and Internationalization Requirements
8.c. Learning Requirements
8.d. Understandability and Politeness Requirements
8.e. Accessibility Requirements
8.f. User Documentation Requirements
8.g. Training Requirements

9. Look and Feel Requirements

9.a. Appearance Requirements
9.b. Style Requirements

10. Operational and Environmental Requirements

10.a. Expected Physical Environment
10.b. Requirements for Interfacing with Adjacent Systems
10.c. Productization Requirements
10.d. Release Requirements

11. Cultural and Political Requirements

11.a. Cultural Requirements
11.b. Political Requirements

12. Legal Requirements

12.a. Compliance Requirements
12.b. Standards Requirements

III. Design

1. System Design

1.a. Design goals

2. Current Software Architecture

3. Proposed Software Architecture

3.a. Overview
3.b. Class Diagrams
3.c. Dynamic Model
3.d. Subsystem Decomposition
3.e. Hardware / software mapping
3.f. Data Dictionary
3.g. Persistent Data management
3.h. Access control and security
3.i. Global software control
3.j. Boundary conditions

4. Subsystem services

5. User Interface

6. Object Design

    6.a. Object Design trade-offs
    6.b. Interface Documentation guidelines
    6.c. Packages
    6.d. Class Interfaces

IV. Test Plans

1. Features to be tested / not to be tested

2. Pass/Fail Criteria

3. Approach

4. Suspension and resumption

5. Testing materials (hardware / software requirements)

6. Test cases

7. Testing schedule

V. Project Issues

1. Open Issues

2. Off-the-Shelf Solutions

    2.a. Ready-Made Products
    2.b. Reusable Components
    2.c. Products That Can Be Copied

3. New Problems

    3.a. Effects on the Current Environment
    3.b. Effects on the Installed Systems
    3.c. Potential User Problems
    3.d. Limitations in the Anticipated Implementation Environment That May Inhibit the New Product
    3.e. Follow-Up Problems

4. Tasks

    4.a. Project Planning
    4.b. Planning of the Development Phases

5. Migration to the New Product

5.a. Requirements for Migration to the New Product
5.b. Data That Has to Be Modified or Translated for the New System

6. Risks

7. Costs

8. Waiting Room

9. Ideas for Solutions

10. Project Retrospective

VI. Glossary

VII. References / Bibliography

VIII. Index

I Project Description

1 Project Overview

Era-set Boulevard is a step towards helping break the stigma on history, a boring often repetitive series of events across the world littered with dates and times of long-winded named events. Through an interactive interface coupled with location based historical events waiting to be told by AI of people of the era, learning history through books is a thing of the past.

2 The Purpose of the Project

2.a The User Business or Background of the Project Effort

Our business is to revolutionize the way that history is learned, by pulling from all avenues of historical resources and compiling them into cohesive interactions with virtual AI in places across the world in which these historical resources reference. This allows us to give a more thorough and unbiased opinion through a vast amount of historical resources and give a more wholesome learning experience when discovering the past, all the way up to the modern era we live in.

The ultimate goal is not to do away with history books, or completely renounce the teachings found standard in schools across the world, but instead to act as a supplement to historical references in every region and act as a motivator to draw people into exploring the world around them, as well as regions oceans away, all while enjoying a knowledge rich experience.

The problem at hand has been that historical references from various regions are often found to have a varying degree of bias and tend to follow numerous unspoken rules, such as “Them who has the gold makes the rules.” or “The victor writes history.” By bringing an enormous amount of historical references to the table, we are able to bring the bias of the content down to almost nothing, and in the case we are not able to fulfill that, we are able to represent all facets of the event and allow users to come to a more wholesome conclusion on events.

2.b Goals of the Project

This project is being developed to provide a supplementary tool to learn about history, while attempting to remove the bias from what a user knows currently about the historical events through the eras in the area they utilize the app in.

Specific goals of Era-set Boulevard:

Immerse users in the era that they have specifically chosen, as well as the region they are currently in by altering the terrain of the app based on these prior.

Remove the bias of history in the era and region the user is in currently by compiling as many historical reference from that region and area into a cohesive and unbiased presentation through the virtual AI.

Motivate users to further their knowledge by having a leveling system combined with titles, the last title being “Historian”, which is coupled by the region and era the user has focused the most on. Ex. “American Colonial Historian”

Bring attention to famous figures in history that have made a significant impact on their region and area by having them represent their region and area as the virtual AI’s form. Ex. Abraham Lincoln at Gettysburg National Military park.

Allow users to interact with famous objects of history from their region and area when they fulfill certain requirements and unlock the achievements that provide these objects as rewards.

2.c Measurement

Measurement of success for Era-set Boulevard will be measured by online users per region, total online users, and user base as a whole.

Our aim is to have at least 5% of the population in each habitable, technology available region that has access to the app, is using the app.

3 The Scope of the Work

3.a The Current Situation

The app itself is an entirely novel idea, making it the pace setter for any, if any, apps of this type come after it. The largest issue is the massive amount of information that must be placed with each of these AI, and how it should be organized.

3.b The Context of the Work

In order to ensure that the app itself works without flaw and to its expected experience, the servers, in which the information for the AI will be hosted on, as well as the player information, need to be as optimal as it comes. Research must also be done in terms of the best spread and quantity of servers per the spread between them in order to ensure the best connection.

3.c Work Partitioning

Business Event List

Event Name	Input and Output	Summary
1. Landmark	click on the landmark icon(in) description/ quiz/ edit icon/points(out)	once click on landmark icon, it will display description of landmark, give ability to click on quiz , edit information , give the user points.
2.Quiz	click on quiz icon(in) increase point(out)	allow user to answer several question about the specific landmark to gain additional points
3.Map	moving around(in) update map(out)	when user move around , the map will be update and display new landmark nearby
4.Login	username/password (in) gain access to map(out)	give user to create or access to their account , to be able to play the game.

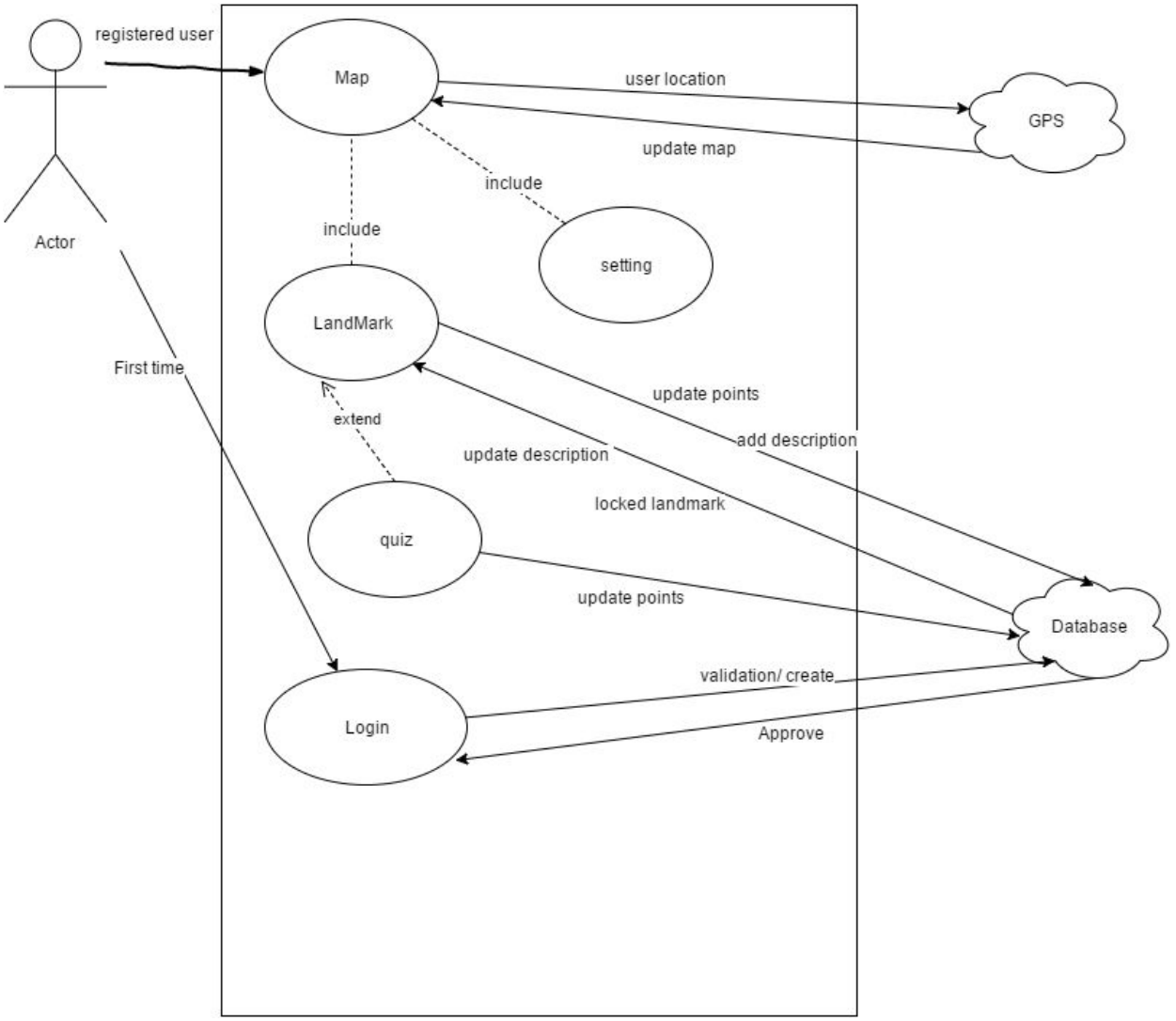
3.d Competing Products

Currently no competing alternatives exist, but the learning of history is core in all regions of the world. Because of this, the potential audience for Era-set Boulevard is massive and because there are no alternatives, we can develop a more wholesome experience without as heavy a focus on competition.

4 The Scope of the Product

This project is simply an educational game. It's not supposed to be something super-secure or have perfect networking. Most of the gameplay itself is handled offline with server communication kept at the minimum. Thus the scope is not as big as it may seem like. The most complicated part of the project networking-wise would pretty much be the hiscore as a server would be needed to get hiscores from everyone. Adding anti-cheating stuff shouldn't be that high of a priority since it's just an educational game above all else. It's not supposed to be something competitive.

4.a Scenario Diagram(s)



4.b Product Scenario List

1. User starts the game
2. User is asked to register via facebook or other means.
3. Once logged in, the app should be at the main menu.
4. Pressing “play’ will start the game.
5. Visually the game will start off like Pokemon GO with your position represented as an arrow, and landmarks represented by other various icons.
6. The user then has to physically go near the land marks and press the icons once close enough.
7. A menu will appear with an image that references the historical significance of the landmark and text that describes what occurred there.
8. Once done the player will get some points, then he has the option of doing a small quiz for additional points.

9. The player then closes out of the landmark screen and goes back to the main game, and the landmark can't be activated for a small amount of time.

4.c Individual Product Scenarios

1. The user opens the app by selecting it from their phone as they would with any other app. Initially the app should go to a splash screen while all assets are being loaded into the app. Once the assets are finished loading it will move on to the next screen.
2. The app will then try to detect the user's google account or facebook and see if it is registered. If the user is not registered then it will ask the user to register, if the user refuses to register then the game will be played on an "offline account", meaning the entirety of the account will be saved on the phone's memory. If the user is registered then it will skip this step and go straight to the main menu
3. Once all the registration shenanigans is over, the app should be in the main menu with three options: play, settings, and exit. The latter two are self-explanatory and can be designed however you want, but we'll be focusing on the play button.
4. Once "play" is pressed, then the game initializes and a loading bar is shown as the app tries to generate the map, land mark data, etc. Basically just initialize the game on a loading screen.
5. Once done loading, a visual similar to Pokemon GO is shown. As in your position is represented by an arrow sign pointed at your position in a GPS map with landmarks being represented as icons on the GPS map. This is supposed to be visually similar to Pokemon GO, so if you don't understand this description of the visual then go either play Pokemon GO or look at videos of it.
6. Then to get points, the user will have to physically go to landmarks on their map. Once they are near a landmark, they can "activate" it by tapping the icon on their screen. The distance threshold that will let the player "activate" landmarks should be rather generous.
7. Once the landmark is activated, a screen will pop up with an image that is related to the landmark and some text on its historic significance. The text can be paragraphs or summaries.
8. Once the user hits "next", he will gain some points and a dialogue box will appear asking if the user wants to take a quiz for more points. If activated a the user is asked 3 randomly generated questions and receives points based on how many he scores correctly.
9. Once finished, the landmark will be disabled for a set amount of time so players have to keep moving and going to different landmarks to gain more points. This would be some sort of anti camping mechanism.

5 Stakeholders

5.a The Client

This product is for the general population. Any individual of any age may find this product to be of interest to them and is free to use. The clients in for this product shall be a company seeking a game that involves managing a multiplayer mobile-app looking for sale in in-game micro-transactions.

5.b The Customer

This product is intended to be focused onto customers that have interest in history, travelling, and socialization. Users of this product will be users seeking a different genre of a location-based game that emphasizes on geographic locations for the purposes of in-game progression. The customer who may also find interest in this product are those who want to experience the location-based game with an intention to learn more about their immediate vicinity. Those who wish to download the program have intentions to play the game by the rules, and only paying when they prefer to experience other and additional features that serve as a convenience to the user.

5.c Hands-On Users of the Product

The hands-on users of this product are all consumers. However so, the consumers can be generally split into the following categories:

- **Historians/Teachers:** A user under this category would still be considered a consumer, however their role in the product's usage is different. The historians may contribute to the game's development by requesting and adding additional historical sites into the game's database, as well as add descriptions to the historical spots with the game developers consent. Teachers on the other hand, may use this product to educate students and their peers with the information provided from using this product. Additionally the people in this category generally provide guidance to their peers as well as promoting safety guidelines for those who play. It's quite possible that users in this category are expected to purchase in-game features that allow greater control over the game.

- **Students:** The users under this category are generally faced with tasks provided from teachers who also use this product. A student may use this product only solely for the purposes of education. This can be done in forms of assignments given by the teacher.

- **General Consumer:** Users under this category intend to use the product for self-entertainment. This category is this product's greatest projection of traffic and target audience. Since there will be most of the users in this category, it is expected that general consumers are also going to be the bulk of the server usage

and the bulk of the general progression of this product. Such as beta-testing, bug reporting, user experience/response, reviews, and revenue.

- **Reviewers:** Small in numbers, but these users will ultimately do cover reviews, gameplay reviews, and performance review about our product. Reviewers provides us a clearer connection to the general consumers; we may receive the current, largest, or important key aspects of the product that may be impacting the quality of our product.

- **Hackers:** It is important to pay attention to these users as they are the ones who ultimately break through the security systems of our product in order to obtain what they seek for in our product. Hackers may disable our servers, hack other players (or personal info), or sabotage the economy of the product. Otherwise hackers will remain a threat to the game, and it will be our product developers to focus on a hacker's method of breaching the product.

5.d Priorities Assigned to Users

Although our product is open to the entire audience, each particular group of users may have different projected impacts on the growth of this product. Categories are listed below:

Key Users: General Consumers (see above section), is the target audience this product is aiming for. We hope to find users in large numbers that can test the product and review the product. Ultimately, the quantity of users in this group is what allow us to gather information about the performance about our product, and the way to access to adaptations that can be necessary to the development of this product.

Secondary Users: Teachers/Historians, Students, Reviewers, Hackers. Secondary Users are predicted that they are to use our product in short durations rather than a long-term user. However our product developers may pay attention to the voice of these users due to their contribution to the development of our product. For hackers, they are important due to their ability to breach the servers, thus it allows the game to grow more secure when work is put forth to defend against hacker attacks. Thus providing additional security to our product.

Unimportant Users: Users who do not own mobile devices are generally void of our product and services until they obtain a mobile device.

5.e User Participation

As this product is a multiplayer game, it is necessary the quantity of users who use our product remain steadily high. This way, we may receive feedback from the users on many aspect of the games such as the design of the GUI, bugs, and/or gameplay.

However there is no specifically required goal for the users of this product to achieve, users are encouraged to provide feedback and reviews when using our product.

5.f Maintenance Users and Service Technicians

See above. Reiterating, hackers and reviewers are crucial to the development of the game. Hackers can provide ways of breaching our product, and our goal is to work against those methods. If the hackers are white hat hackers, they can work for us to find ways to breach our product with consent. Reviewers and our own technician teams can cooperate in producing a well refined product.

5.g Other Stakeholders

In addition to stakeholders mentioned above, these users may contribute to the development of our product. These are the users as follows:

Art/Graphic Designers: May provide feedback on the design of the game, and can be hired if their services are deemed necessary and beneficial to the health of our product.

Development Teams: Not from our corporation ourselves, but from other organizations. These teams may assist in our product development, or may provide suggestions.

Federal Investigation: Although seldom, users in this category will be highly trained investigators who may use our product to investigate legal issues that may occur within or outside of our product's scope. They may use this product to investigate GPS, locations, times, and other conditions that are important to investigation.

6 Mandated Constraints

6.a Solution Constraints

Description: This game will run on mobile android devices, this includes phones and tablets.

Rationale: Since the game is a GPS-based location game it means we have to primarily focus on devices which have GPS devices like phones and tablets. Due to this, Desktops, Laptops, etc will not be included.

Fit Criterion: The product will run normally on all android phones and tablets. No initial apple support until the android-side is stable.

6.b Implementation Environment of the Current System

As described previously the environment of this game will primarily be android phones and tablets, all working on 4.0 Ice Cream Sandwich as the minimum require. Ideally the game will work equally on both tablets and phones with the only

difference being resolution stuff, so the screen is bigger. No apple support as stated previously, because it would over-complicate the project, it'd be better to tackle things one at a time.

6.c Partner or Collaborative Applications

This game will mainly be a solo game so there is no "collaborative applications" gameplay wise. But development wise this should be done in collaboration. Having a separate person maintaining each front-end of the product would be ideal so we can split up and tackle tablets, phones, etc, separately/

6.d Off-the-Shelf Software

This will be mainly be made in Android studio. Using a library that will help in coding the game is fine such as LibGDX. Another library may need to be used to map the GPS to your location. If there is extra development time, Kryonet could be used to handle all the network coding because it simplifies all network-related code. Git would be needed to work in a group.

6.e Anticipated Workplace Environment

The workspace for this shouldn't be anything too complicated since this is just a school project. Simply having all members be in constant communication via a chatting application should be fine. Having occasional meetings after class should be enough as long as work is separated equally.

6.f Schedule Constraints

Not applicable, turn it in on time I guess?

6.g Budget Constraints

Not applicable.

7 Naming Conventions and Definitions

7.a Definitions of Key Terms

Historical Landmarks (Landmark): These refer to the landmarks spotted in the game which correlates to the actual landmarks in the real world, as well as the landmarks in the game user interface. The landmarks provide *EXP* and are accessed through physically reaching the location. Some landmarks may have *quizzes.*

Quiz: Found in some *historical landmarks*. These provide bonus *EXP* by user's consent when they accept and take the quiz. There is no penalty to failing the quiz.

LVL/RANK: Stands for level, as in Player level. Starts at 1, goes up to 40. Each level may have their own nickname on that rank. The levels are a representation of the player's progression on the game, where higher levels means more unlocks and perks the player will get.

EXP: Stands for experience points, these points are earned through investigating *Historical Landmarks*. These points are accumulated by the user, and are used towards the user's *Level*.

GOLD: Refers to the in game currency that the user may use to spend on premiums or conveniences. Earned through payment with U.S. dollars or other in-game events such as *Leveling up* or completing a daily task.

7.b UML and Other Notation Used in This Document

(Not applicable)

7.c Data Dictionary for Any Included Models

Relevant Models:

EXP + Landmarks = LVL + GOLD.

8 Relevant Facts and Assumptions

8.a Facts

- Users will be required to be over the age of 13.

- The application will have more than 7000 lines of code.

- The application will run on the Android devices, as well as requiring access to the internet and the GPS to communicate with the user interface to display landmarks, position, and update the map. Without permission or access to the GPS and/or internet, the user will not be able to use the product as its intended use.

- The user is fully aware that data usages will be charged by their mobile carrier provider.

8.b Assumptions

- The developers of this product will assume that the users of this product will have a stable Android device with GPS and internet.

- Users are assumed to have some experience with navigating the interface.

- Development of the application will take place on the operating system such as Windows, Linux, Apple OS environments as long the developer has access to Android Studios.

- The descriptions and quiz will be open source to the public so that the users have ability to change the landmark description, and make up some new quizzes. Those will be included in the games based on users voting system.

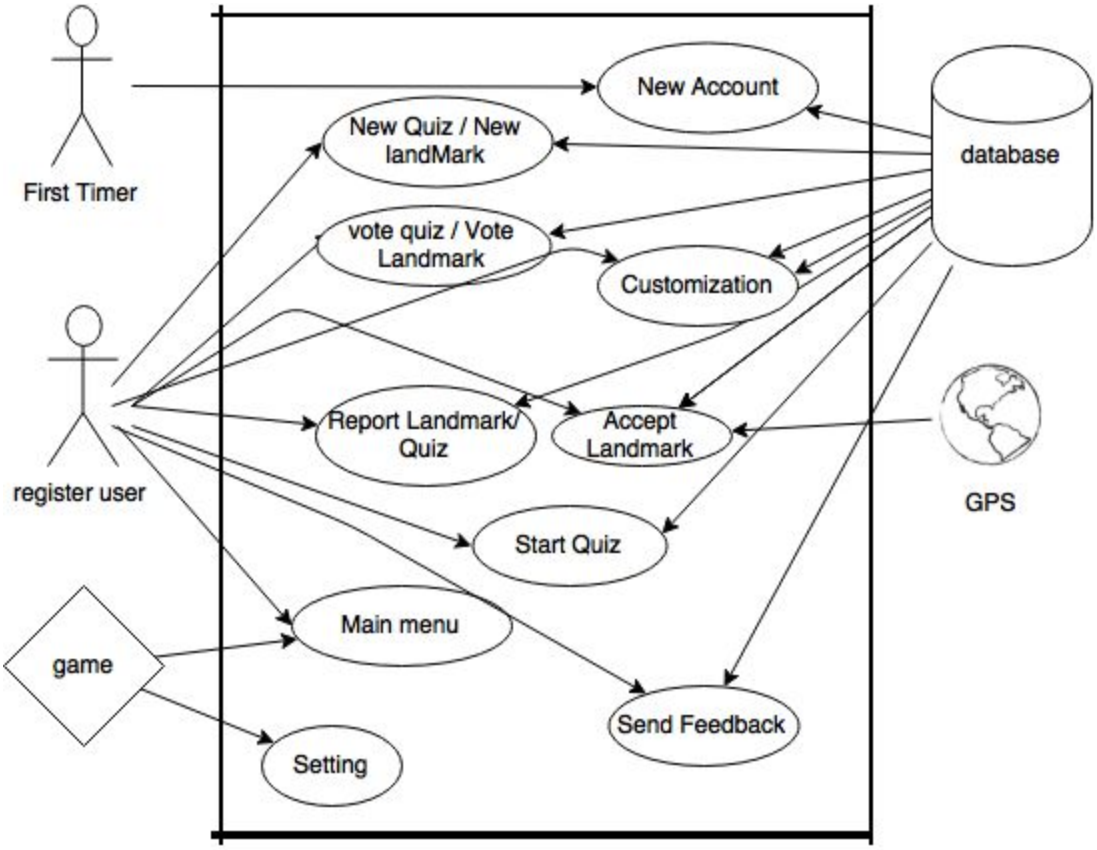
- The application will be updated as needed, such as for optimization, security, functionality, and to minimize any bugs detected by the developers or users.

- Not all locations will contain the same density of historical landmarks.

II Requirements

1 Product Use Cases

1.a Use Case Diagrams



1.b Product Use Case List

- New Account
- New Quiz
- Vote for Quiz
- Start Quiz
- Login
- Main Menu
- Exit
- Settings
- Send Feedback
- New Landmark
- Vote for Landmark
- Accept Landmark
- Report Landmark/Quiz

- Customization

1.c Individual Product Use Cases

Use case name: New Account

Actors: Users, Database

Flow of Events:

New user enters new account, and is prompted for both a user name and a password. Should a user name be taken, the system will alert the user and cycle back to the previous login screen. Once the account has been established, they will then be brought back to the login screen and will be notified of successful account creation. They will then enter the login information required to access their account. Once their account is loaded, the system will bring them to the tutorial to explain all aspects of the application.

Entry conditions: User has moderate skill in mobile applications and mobile internet access.

Exit conditions: Account with login data will be established.

Quality restrictions: N/A

Use case name: New Quiz

Actors: Users, Database

Flow of Events:

User heads to a landmark and prompts to create a new quiz for the landmark. A form displaying a set of five textboxes for entering the question, five sets of four check boxes underneath them with text boxes next to them for each of the answers to the question to be entered and five sets of four check boxes next to them to choose the correct answers. Once the form is filled out, the user clicks the submit quiz button at the bottom of the form and is returned back to the main map.

Entry conditions: User must be of level 5 or higher to submit quiz. User has significant knowledge of the landmark. User is positioned at landmark.

Exit conditions: New quiz is created and stored in the database.

Quality restrictions: Make sure account is logged in and add this question to database of questions to be voted.

Use case name: Vote For Quiz

Actors: Users, Database

Flow of Events:

User heads to a landmark and prompts to see quizzes submitted for the landmark. A list of quizzes starting from the highest rated quiz that has not been accepted to the lowest rated quiz. The user chooses the quiz they would like to see and a form showing all the questions and answers to them appears. At the bottom of the form is two buttons one with a thumbs up and a thumbs down, along with a text box underneath for mandatory comments if a thumbs down is chosen, and optional if a thumbs up is chosen. Once the user has chosen their stance on the quiz, they hit the submit button under the comment text box and they are moved back to the quiz list with the quiz they voted on hidden from their view.

Entry conditions: User has significant knowledge of the landmark, User is positioned at landmark

Exit conditions: Quiz is voted on and feedback submitted.

Quality restrictions: Check if account is logged in to vote.

Use case name: Start Quiz

Actors: Users, Database

Flow of Events:

User heads to a landmark and prompts to start a quiz for the landmark. A form showing one of the accepted quizzes stored in the database is displayed. The user reads and answers all 5 questions and then submits them. The users answers are checked against the correct answers and the amount they got correct is displayed. Based on the amount correct, the user is awarded experience, the amount shown under the amount of questions they have correct. Once the user is done viewing the results, they hit exit, and are brought back to the main map at the landmark.

Entry conditions: User is positioned at landmark

Exit conditions: Quiz answers are submitted.

Quality restrictions: Refer to database of quiz questions to choose.

Use Case Name: Login

Actors: Users, Database

Flow of Events:

When a user launches the game, they will be attempted to be logged in. If this is a new account, the "Register Account" use case will occur. Otherwise if the game detects in their history that they have a registered account and wishes to be automatically logged in, then no prompt will appear and the user will be logged in and brought to the main menu directly. Otherwise if the user wishes to log-in manually every time via user preferences, then he will be prompted to log in. If the user is playing in offline mode then no login prompt will appear.

Entry conditions: User Launches the game and has an account

Exit Conditions: Login was successful or no login if in offline mode

Quality Restrictions: Save login preferences in the users config and use that.

Use Case Name: Main Menu

Actor: Menu Handler

Flow of Events: Once the login/registration process is over, the user will be lead to the main menu screen. In this screen all settings, etc, must be initialized from the user's preferences. There will be several buttons, such as play, settings, exit, etc. If the user hits play then the main menu will exit and the game begins. For the other options they will do what their names imply, thus pressing quit will quit the game, pressing options will open the options menu, etc.

Entry conditions: User has successfully authenticated/is in offline mode

Exit Conditions: N/A

Quality Restrictions: Refer to the user's screen size to scale the menu if necessary.

Use Case Name: Exit

Actors: Game

Flow of Events:

Once the user wishes to exit the game entirely, all memory related stuff must be executed. Such as freeing memory, saving settings, etc. Once done then the game can be successfully “exited”. If the user presses quit from the menu, then a confirmation dialogue needs to be added in just incase the user pressed it by accident.

Entry conditions: User presses exit on the main menu or exits the app entirely.

Exit Conditions: Memory management was successful and any game related variables were saved.

Quality Restrictions: It’d be helpful to have a gamestate tracking class to see what needs to be done based on how the program is running.

Use Case Name: Settings

Actors: Config

Flow of Events:

When a user launches the options menu a list of settings will be displayed. This will include things such as render detail, offline mode, etc. If a user changes a setting then the game will automatically save it to the users preferences file AFTER he exits the settings menu so all the changes can be processed at once. The preference file should be saved locally on the users device, meaning his preferences will not transfer over if they are logged in on a different device. Hitting apply will save the settings, exiting the options menu without hitting apply will add a confirmation dialogue asking if they wish to keep or discard the changes they just did to their settings. Additionally a “default settings” option should be added incase the user messes up his settings by accident.

Entry conditions: The user opened the options menu

Exit Conditions: Setting were saved properly if anything was changed, otherwise do nothing.

Quality Restrictions: Using a third-party library to handle user preferences in a config file should make this a lot easier.

Use Case Name: Send Feedback

Actors: Users, Database

Flow of Events:

Upon the user landing on the main menu screen or in game screen, the user must access the settings. In the settings the user can send feedback. In the feedback form, the user can name the feedback with a title, categorized combo box (i.e. suggestion, bug, complement, etc), and their description. After sending it, the feedback is sent onto the program's servers for the developers to read. Or often enough, the program will automatically prompt the user for a feedback, in which it's optional to send and can be skipped.

Entry conditions: The user is viewing the settings screen. There may be a level requirement.

Exit Conditions: Sends a feedback to the system.

Quality Restrictions: N/A

Use Case Name: New Landmark

Actors: Users , Database

Flow of Events:

User head to the new location and submit a photos , text , gps location , Description to that new landmark , once that information filled, press submit button to end information form. resume back on the map

Entry conditions: user at the new location and have no landmark icon on the map

Exit conditions: New Landmark is created and stored in the database

Quality Restrictions: Make sure the user have account and add the new Landmark to the database to be vote

Use Case Name: Vote for Landmark

Actors: Users, Database

Flow of Events: Users selected vote for Landmark on the main map, it display all landmark that is submitted from highest rated to lowest rated landmark that have not been accept . when user click on the landmark , it display description of the selected landmark and have two button for thumb up and thumb down , once user selected the button , it will go back to the main map.

Entry conditions: user selected the vote for landmark on the main map

Exit conditions: landmark have been voted

Quality Restrictions: record the vote back to database

Use Case Name: Accept Landmark

Actors: Users , Databases , GPS

Flow of Events:

Once the User click on accept Landmark button , The description , the name and the picture pop up on the screen , with buttons lead to Quiz, create quiz , vote quiz , back to main map

Entry conditions: Landmark icon existed on that map , The users is in that location,

Exit conditions: Depend on users selecting the button , should lead to Quiz , create quiz , vote quiz , back to main map

Quality Restrictions: N/A

Use Case Name: Report Landmark/Quiz

Actors: Users, Database

Flow of Events:

Upon viewing a historical landmark, should the landmark be faulty or that it features inappropriate conduct (such as fake description, profanity, fake location, etc), then the user may file a report using the red flag button. The report has a few category checkboxes that label the report, and a brief description of the report. After enough reports with similar cases, the landmark or quiz will be removed accordingly by the system. The user tied to creating the said quiz or landmark will either be banned or suffer penalties/ warnings.

Entry conditions: The user is viewing a landmark.

Exit Conditions: Sends a report to the system.

Quality Restrictions: The user cannot see who they're reporting, only the system knows who created the landmark.

Use Case Name: Customization

Actors: Users, Database

Flow of Events:

When viewing the open world/map screen, the user may select their character avatar. Here the user may customize the appearance of their character according to some general criteria: hats, hair color+style, gender, shirt, pants, shoes, accesories, and other physical properties. The user is given some default avatar, while other avatars are received through events, leveling up, or in game purchases.

Entry conditions: The user must be in the map screen, then click on their profile avatar. The user then shall have to have avatars if they wish to change their defaults.

Exit Conditions: The avatar appearance is saved onto the database.

Quality Restrictions: Avatars do not affect the user's performance in the game whatsoever, only for cosmetical reasons.

2 Functional Requirements

Requirement: User must be automatically logged-in if his preferences say so.

If the user enabled automatic login from either the options, when he first registers, or whenever he logs in, then the game must stop prompting the user to log in and automatically do it for him using facebook, the built-in accounts, google plus, etc. The game should ask the player to log in once a month anyways just to make sure the user is still in control of his account.

Fit Criterion: Like other games/websites just have a tic-box everytime the user logs in that says “automatically log me in” and the user can opt-in if he wants.

Requirement: A menu system must be made for ease of use

Basically just have a simple menu system and don’t overcomplicate it for ease of use. If possible, use a library that does the work for you or alternatively just use the main menu as a template for all other menus.

Fit Criterion: Use a template that you’ll follow every time, having something drawn out as a reference may be helpful.

Requirement: A database needs to be made to hold all the quiz questions

All user-submitted questions need to be stored safely. Using mySQL for this would be ideal because then we can add and remove questions on the fly using the database. Using mySQL specifically would be ideal so a lot of “grunt work” is done for you and your efforts can be shifted elsewhere. Of course you’ll need to know how SQL works and make sure it turns efficiently and there are no stalls.

Fit Criterion: Integrate sql into the app.

Requirement: User gps location must be display on the main map

User location is a must have to interact with many different landmark , answer quiz , read description about the landmark and gain EXP for every visit to landmark and answer the quiz.

Fit Criterion: Gps location must be display at all time when user is using the app

Requirement: The internet or the Databases must contains landmarks and it’s description .

The main fun of the games is to learn about the landmarks around the users, it must be able to get the information of the landmark quickly as possible and minimize the loading time to get the information

Fit Criterion: must be fast and reliable , focus primarily on reliable .

Requirement: the Tracker of the user experience points

Must be able to keep track of all the experience points user have accumulated throughout the game. That way user will be able to unlock benefits and perk while they playing the game.

Fit Criterion: must be balance for the rewards and accurate counting of the experience points.

Requirement: The program must be able to run with Android 8.0 and higher, or iOS 9.0 and higher.

Since most functionality in form of GPS and mobile data plans consists of operating systems in this criteria, then the program will likely need the operating systems as specified.

Fit Criterion: Downloading should still work on operating systems older than the ones listed.

Requirement: The user must be notified if a update available is pending.

It is important that if the user wants the most recent and refined experience of our program, then they should update at the earliest opportunity. Since there could be bugs that could potentially ruin the experience for the user, the updates strive to fix it.

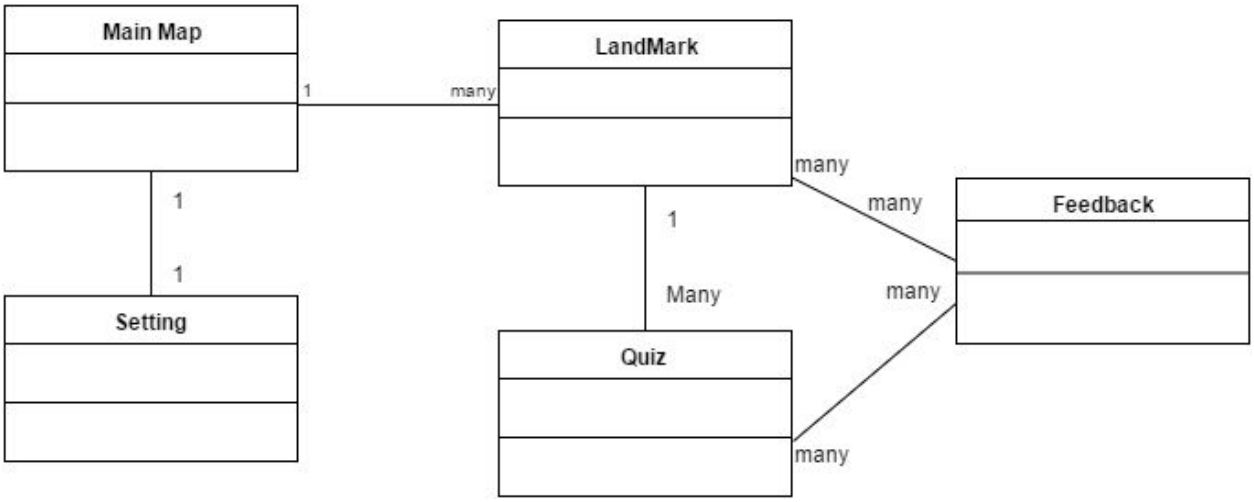
Fit Criterion: The update shall describe what is changed.

Requirement: Any changes to the terms and agreement must be notified to all users.

It is important to note that at anytime, the services of our program may change such as the engine, API, or other sources which come with their own terms, the user may want to know what has changed, and whether or not if they agree or disagree with the changes.

Fit Criterion: The user must either accept or decline these, changes. Accepting allows continued use, and declining will reject program use.

3 Data Requirements



4 Performance Requirements

4.a Speed and Latency Requirements

Login should take no longer than 30 seconds to match username and password from database.

Creation of account should take no longer than 15 seconds.

Movement update on minimap should occur at worst up to three seconds.

Submission of new quizzes should take no longer than 30 seconds.

Voting on quizzes should take no longer than 15 seconds.

Submission of quiz answers and return of result should take no longer than 30 seconds.

Accessing account data: level, experience, currency, quizzes done, etc. should be near instant by keeping a set of local variables on client side to be updated based on the server in order to compensate for possible cheating.

Precision or Accuracy Requirements

Everything in the system will work based on seconds, with exception of account data access and anything else that relies solely on client side interaction.

4.b Capacity Requirements

The server will be capable of supporting tens of thousands of users on the server at the same time.

The database and its queries will allow access to all account information for every user on the server and will keep them stored for users that have been active within the last six months.

The database and its queries will allow access to all current and future landmark information permanently.

The database and its queries will allow access to all accepted quizzes permanently.

The database and its queries will allow access to all submitted quizzes that are voted on, within the last six months.

5 Dependability Requirements

5.a Reliability Requirements

On the most optimal and stable conditions, this product shall not stop involuntarily. A definition of a case when this product may fail to function properly is a state when the program is unable receive any input from the user as well as produce any output either. Cases when this situation happens generally requires the user to shutdown the software externally and restart as desired. The program shall continue to work as usual afterwards.

In the case when the software crashes and cannot continue usage, the program will exit without any memory leaks, as known as a safe exit. There are a couple cases of how this scenario will play out. In the most stable exit, the program will have a in-program message prompting the user that the program has dealt with an issue and requires that the user must restart the software which is shown with a exit button that the user must voluntarily press. The less stable exit is when the program exits without warning and afterwards there will be displayed a message stating that the program failed due to “x” reasons, for debugging.

For the case when the software faces a fault, or illegal memory access, then the program will first warn the user that an error has occurred, and whatever transaction the user was doing was/may be lost or not been completed, and the user must retry their request again.

Another type of fault the user may encounter is when there is a loss of power or loss of contact to the user’s running system. In this case the program, after the user

restarts, shall notify the user that the program have once crashed unexpectedly, and some data may have been lost. The best way to combat this is by making sure our program makes periodic saves to the server so that not all is lost in a case of a power outage.

5.b Availability Requirements

The product will be available to download and play for at the maximum of 365 days to the year, and 24 hours to the day given that our program does not require any updates and maintenance.

In the case of a program server failure, the program's correlating contact site and the program itself should notify users that a fix is currently in progress, and the program will be ready in no more than 6 hours. If the fix takes longer than 6 hours, then user compensation may be required to fulfil customer satisfaction.

In the case of a program maintenance or update, the user's program should be updated before continuing use. Depending on the type of update, the user may still be able to run the program. In the case of a large update, the program shall notify that the user cannot play until the update has been completed via their application source.

As this product runs on location and internet/data use, the product is expected to fail without those present on the user's side, and will prompt the user that such faults is currently happening that the user must address to in order to re-obtain functionality of the product regarding the fault.

5.c Robustness or Fault-Tolerance Requirements

To make sure the product is available for use for most users, the product shall handle faults on the user's side. So in such, when a fault does occur to a user, the program shall try to continue if the fault is minor and notify the user. Other larger faults may notify the user that something has gone wrong, or error, and note that a user transaction may have been incomplete, or that the program may require restarting. In the case of a server sided error, then notify users that the program is down for maintenance until fixed.

When the program faces a critical error, then the program must know when to automatically close. Whether it automatically closes or not, when the user restarts the program, the user should receive a message noting a critical failure has occurred and some data may have been lost. To boost future failures, the user is then prompted to if they want to send their program failure debug data (which is collected by the program upon crash) for testing.

5.d Safety-Critical Requirements

Content

Although our product cannot directly physically harm the user, we are considering that this program requires the user to go out in the outside environment including the element of distraction and loss of focus due to the user using our program. So the program shall always warn the user not to do a following list (and not limited to) of dangers:

- **Do not Drive and use this application!**
- **Avoid traffic and do not trespass other person's property!**
- **Keep aware of the environment around you.**
- **We recommend not to play at night.**

With general understanding of the dangers of being unaware in an unsafe environment, injury, death, or accidents may occur if the user is not giving their undivided attention to the environment around them.

To try to combat this, the program shall also additionally attempt to generate the historical landmarks in safe locations for the user to go in. In such, historical landmarks shall not be created in locations that also have high traffic, construction sites, and private property.

In the case the user does wander onto private property, our program is not responsible for the charges that may infringe onto the user.

An overlooked situation may include the power consumption this product may use, and prolonged usage of this product may overheat your phone or battery when in a hot environment. There has been very few case in which a few select brand and name of phones that may combust or melt when the battery is overheated. This is in no way related to our product.

6 Maintainability and Supportability Requirements

6.a Maintenance Requirements

In the case of an emergency update, users currently on the server must be updated that the server will shut down in roughly 30 mins from the time the update has been decided.

The application should have between a two week to two month update cycle based on the amount of content updated.

Any updates prior to being pushed must be tested thoroughly to ensure that the least amount of flaws are being put onto the live version in order to ensure a good user experience as well as a good developer experience.

6.b Supportability Requirements

The tutorial can be permanently accessed will be readily available in game, after the user has logged on the first time and completed it.

A forum will be available for users to communicate with each other, the developers, post suggestions, and more.

The forum will have a button that is permanently accessible from in game, allowing users to swap between the application and the forum.

Developers of Era-set Boulevard will serve as the main administrators for the forum along with chosen players.

6.c Adaptability Requirements

The software will be supported on Android and iOS.

Initial software will be english-focused, but will eventually contain support for as many languages that the market can be expanded to support.

6.d Scalability or Extensibility Requirements

The amount of servers we have should expand from being able to handle tens of thousands of users to hundreds of thousands of users in roughly a five year timespan.

The table sizes for the database should be optimized from being able to access and handle tens of thousands of users information to hundreds of thousands of users information in roughly a five year timespan.

6.e Longevity Requirements

The application shall be expected to operate within the maximum maintenance budget for a minimum of five years.

The servers should be expected to operate within the same span of time as the product.

Based on success, the application will aim to double the expected minimum time of operation.

7 Security Requirements

7.a Access Requirements

Data protection is not as important an application like Era-set Boulevard as it is for other applications. Keeping players from obviously editing client side values that can update server values is first and foremost and has been designed around.

Standard methods of encryption will still be used to keep any outside sources from accessing the database, but unless there is a rampant amount of illegal access, not much will be done further except for reactionary updates to deal with small accounts of illegal access to the server.

Only administrators and/or developers, both can be mutually inclusive, may have access to the server.

7.b Integrity Requirements

Database integrity will be protected via digital and physical defenses in order to ensure security and instill a sense of comfort in our users.

Prevention of code injection, gps spoofing, bots, and other various breaches of service will be the starting actions towards protecting data integrity.

7.c Privacy Requirements

User privacy is taken with the utmost seriousness, and our aim is that in the case that we expand towards micro transactions and any other avenues that may require us to hold user information that could possibly be used to expose the user's identity or any private information that can be used maliciously, we will implement a much more enclosed system of protection for our databases.

7.d Audit Requirements

Data auditing will take place solely within physical facilities, this is to keep sensitive information from leaking, regardless of how private the information may be to the user.

No data shall be transported outside of the facilities unless an extreme case warrants this.

Information given by users will only be stored for six months since last user activity.

7.e Immunity Requirements

In order to combat illegal access of the database, a team will perform a string of database attacks over the course of one week every three months.

Copies of the database will be stored on a physical server that can only be access via local usage in case of a critical situation.

The copy of the database stored on the physical server will be updated once every week to ensure safe recovery in case of a critical situation.

Usage of version control software will be mandatory in order to allow for rollbacks on code that is deemed unsafe and exploitable by outside sources.

8 Usability and Humanity Requirements

8.a Ease of Use Requirements

This product is designed for any persons whom are able to have access to a phone under their responsibility and are able to provide self discretion when exploring outdoors. So optimally, we expect that the user is at least 12 years old of age and have a general understanding of the geographical location they are at, and as well as some historical spots around their residence.

As this product is designed for of all interests, we will provide a tutorial for those who needs to learn how this game works. The tutorial will cover over the largest functions our product has to offer and show the user can use them.

Inside the program, it will provide proper and detailed instruction in the Help menu.

Tips will be provided to the user throughout the game. These can be turned off at anytime in the user settings.

For those inquiring any concerns or questions about our product may visit our website or call our number to speak to one of our representatives.

The projected and expected time for when the use should be able to understand how our product works is less than 1 hour.

To make sure our product works for the best of interest of the users using our product, a feedback system will be provided for the users for those who do not feel satisfied with our product or have interests in expressing their concerns with the system.

8.b Personalization and Internationalization Requirements

For it’s the best of interests for our product to comply with the user’s convenience and usability - our product will provide customizable settings.

A few settings is first available to the user before the user can log in. These settings include:

- The ability to mute, or control the sound levels of aspects of the game such as sound effect levels, or music levels.
- Change the language (default English in North America, other defaults would be provided when our product is downloaded in a different region).
- To calibrate the phone, so that it can properly detect your current location.
- Switch and remember/save accounts.

Settings then, after logging on include the color of the map displayed. Enable power saving mode, zooming in/out, character avatar customization, and other visual customizations.

8.c Learning Requirements

The only requirement is that the user should know how to read their own preferred language. Unlike what the theme of the game suggests, the user does not necessarily need to know history to play this game. However it can get a player to progress faster into the game if they knew more.

The expected time for a person to understand the usage of this product is 1 hour, given that they went through the tutorial and played through the game during that time.

8.d Understandability and Politeness Requirements

This software will primarily use historical terms and terminologies to describe landmarks and structures. It is the best of interest for the user to understand these terms when they are used in conjunction with their language of choice (i.e. English).

This means the product may have some terms and words that may offend certain users, and such only historical terms and descriptions are to be used to describe landmarks. However, for user submitted/vote-in descriptions and landmarks, there will be a strict check on the language used. In such, any profanity or misuse of foul words may result in a ban or warning.

8.e Accessibility Requirements

As of current release, the product will adhere to those with eyesight disabilities. For example, users who cannot read but hear, will be able to hear the descriptions of landmarks with sound instead of reading text.

Other requirements may include text font modification, color blindness, and other functions that require attention.

8.f User Documentation Requirements

Most of the internal documentation will be provided in the application hosting app (Google Play, Apple Store, etc). This will include product scope, product usage, images, and videos.

Then an re-iteration of the documentation can also be found within our software. The user must select “About” in the menu tab.

And finally the user may contact our resource center for any questions and concerns.

8.g Training Requirements

There will be no training required for this product. There is however, a instructional video showcasing our product in use, as well as in-game tutorials to help the user start out their game.

9 Look and Feel Requirements

9.a Appearance Requirements

The appearance must appeal to the teen audience and older.

The appearance must cater to the location in which it is situated.

9.b Style Requirements

The design must be simple and straightforward to allow users to pick up the application and immediately be able to navigate.

The map of the user’s location should distinctly represent the era that the user has chosen and the locations features they are currently in.

The interface should distinctly represent the style of the era that the user has chosen.

10 Operational and Environmental Requirements

10.a Expected Physical Environment

The product will primarily operate outdoors and indoors, and ultimately require the user to move between these locations. All locations here are accessible through without this program and thus means that there is no “designated” spot that the game plays in. Since this program is all about exploring, the user is expected to traverse the places by foot or by vehicular transportation.

Some of these places include:

- Parks

- Museums
- National Parks
- Forests
- Schools
- Grassland
- Cities
- Suburbs
- ...and many more.

10.b Requirements for Interfacing with Adjacent Systems

This product must function with smartphones running either Android (>5.0) or iOS (>8.0) operating systems.

These systems must also support the usage of GPS tracking and the user must have mobile data.

Otherwise, until noted, no bluetooth, or any other external devices are required to run this program.

10.c Productization Requirements

This product shall be distributed via popular digital distribution services, such as Google Play and Apple Store. The download and installation will work automatically on these sites.

Unless otherwise stated, our product will not be available anywhere other than Google Play or Apple Store.

10.d Release Requirements

To make sure that the product works in the best interest of our customers, we are to perform periodic maintenance checkups and updates onto the live servers.

Updates will be done periodically to make sure the program can run smoothly and bug-free while providing new content that will promote user traffic with new features, optimizations, and usability.

With the feedback function that the users have, users are able to submit any concerns and any thoughts about the program and the product managers shall access any issues or concerns that are the most appealing.

Must a feature be removed, the product managers shall make a decision to remove a feature in all the best interests of the community and the company.

11 Cultural and Political Requirements

11.a Cultural Requirements

This project is mainly an educational game and thus we do not expect it to affect anyone at a cultural level, especially since the game will be only covering a small area like Skokie or something. We do think that perhaps there may be some discussion on some "touchy" subjects like the trail of tears, etc. But nothing should really happen as long as the history isn't presented in an obviously biased manner.

11.b Political Requirements

Again, since this is an educational game we don't see any political requirements. Since the scope of this project is in USA, and more specifically in Chicago, the game does not have to follow any specific political agenda. But if it were to be expanded into other countries then their political state should be kept in mind when designing questions.

12 Legal Requirements

12.a Compliance Requirements

Adherence to local and federal law shall be enforced to ensure that all aspects of our organization are able to operate in the United States.

Legal documentation to prove our existence as an organization and to receive payments.

All information collected and stored adheres to Data Protection Act.

All original property shall be copyrighted accordingly.

Employees will be bound by a non disclosure agreement as to protect all original property.

12.b Standards Requirements

Employees will follow a 40 hour work week that will consist between at home and at office, a minimum of at least a half of the hours in office.

Employees that work out of office must submit a showing of their work to their managers in a fifteen minute period at the end of the week to ensure that all tasks are being met in a timely and precise manner.

All code shall be commented and documented properly in order to

III Design

1 System Design

1.a Design goals

Design goals of our product will focus heavily on the UI and environment map displayed when traveling from area to area. The UI will be simple and inviting, with soft eye-catching colors that should keep the user from being intimidated by an unknown UI. The environment will imitate colors of the area of the world the person is in, and in the early launch of the game, focus on only showing flat area around roads, before expanding to natural land features.

The transition between environments will be as fast as possible, in order to ensure the best user experience, limited only by the gps and wi-fi the user may have. The UI will also be self explanatory and non-cluttered, allowing users to have a smooth and enjoyable experience.

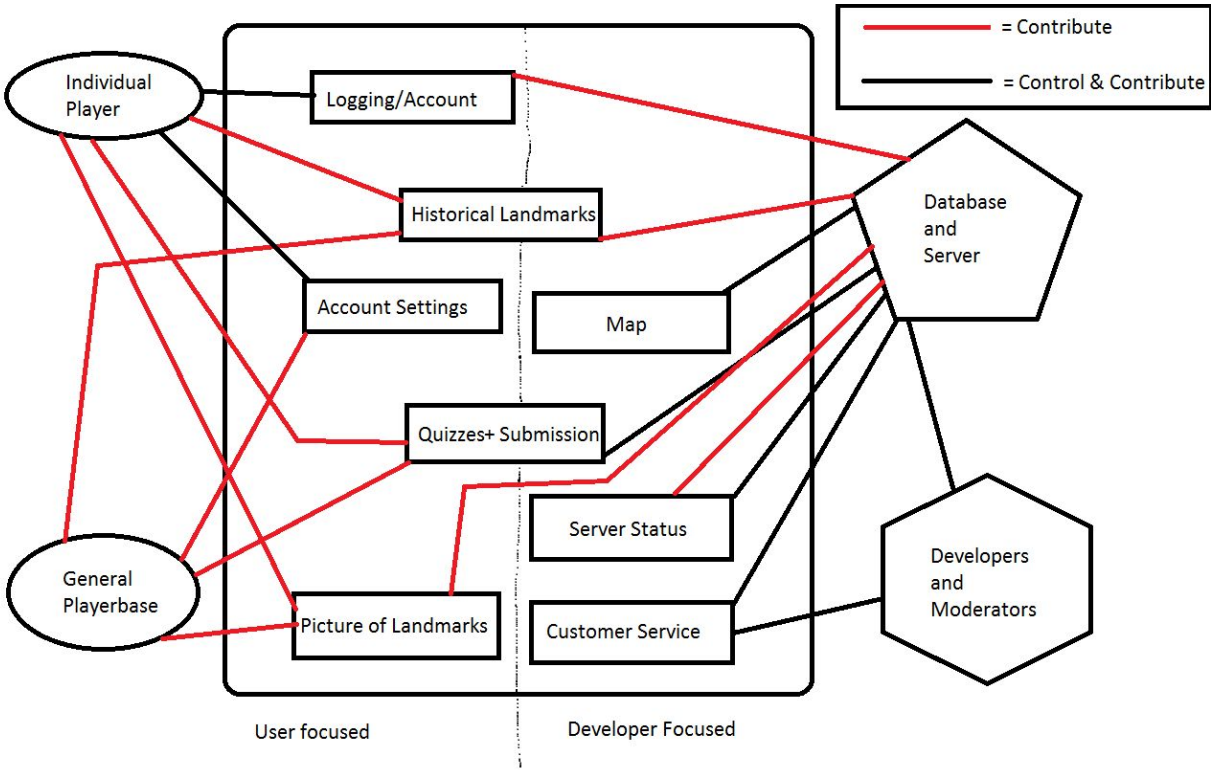
2 Current Software Architecture

Software architecture is defined by the interactions between the entities inside and outside of the source code.

3 Proposed Software Architecture

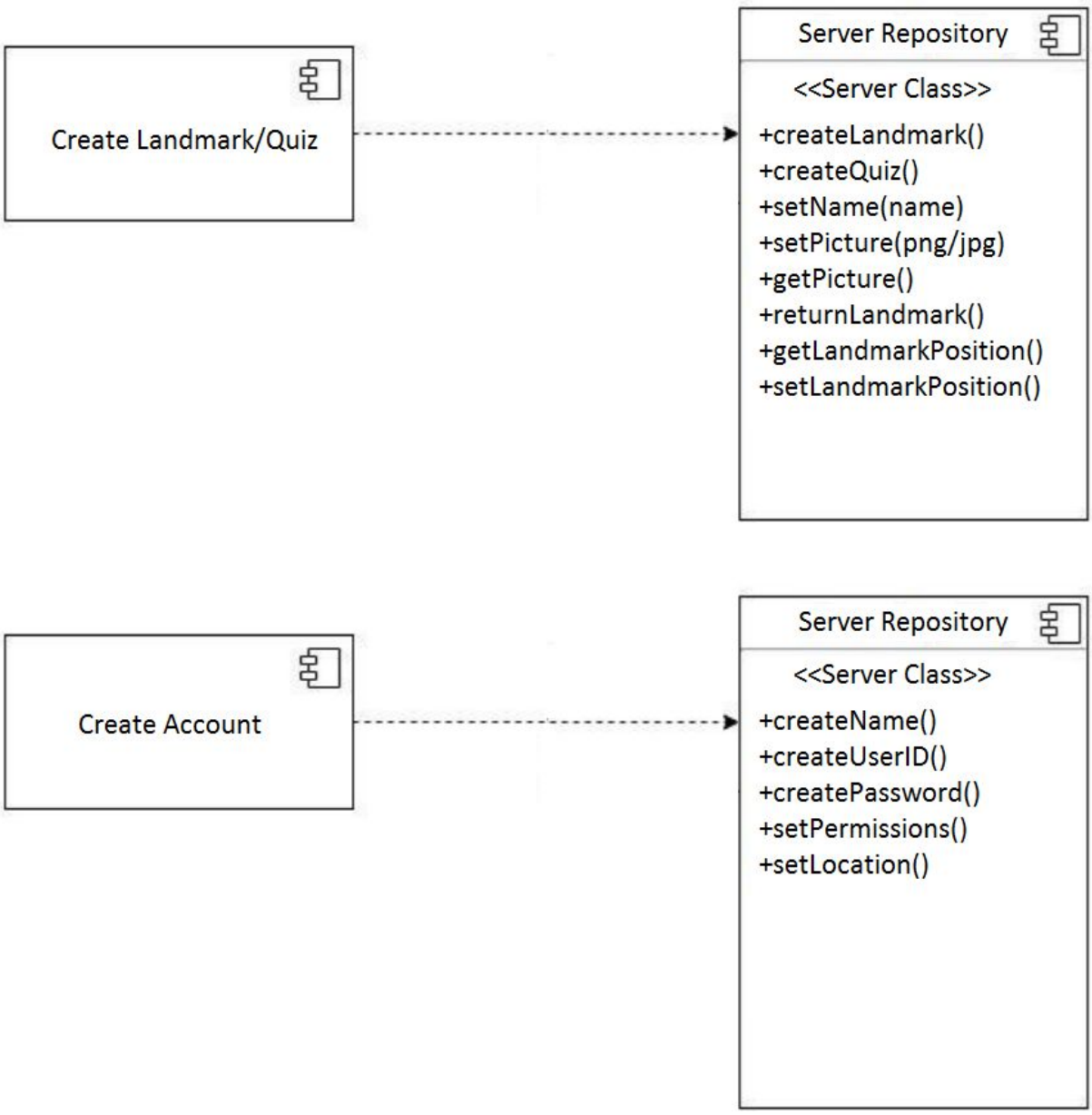
3.a Overview

The following illustrates the general architecture of the game with its functions and contributions/contributors.



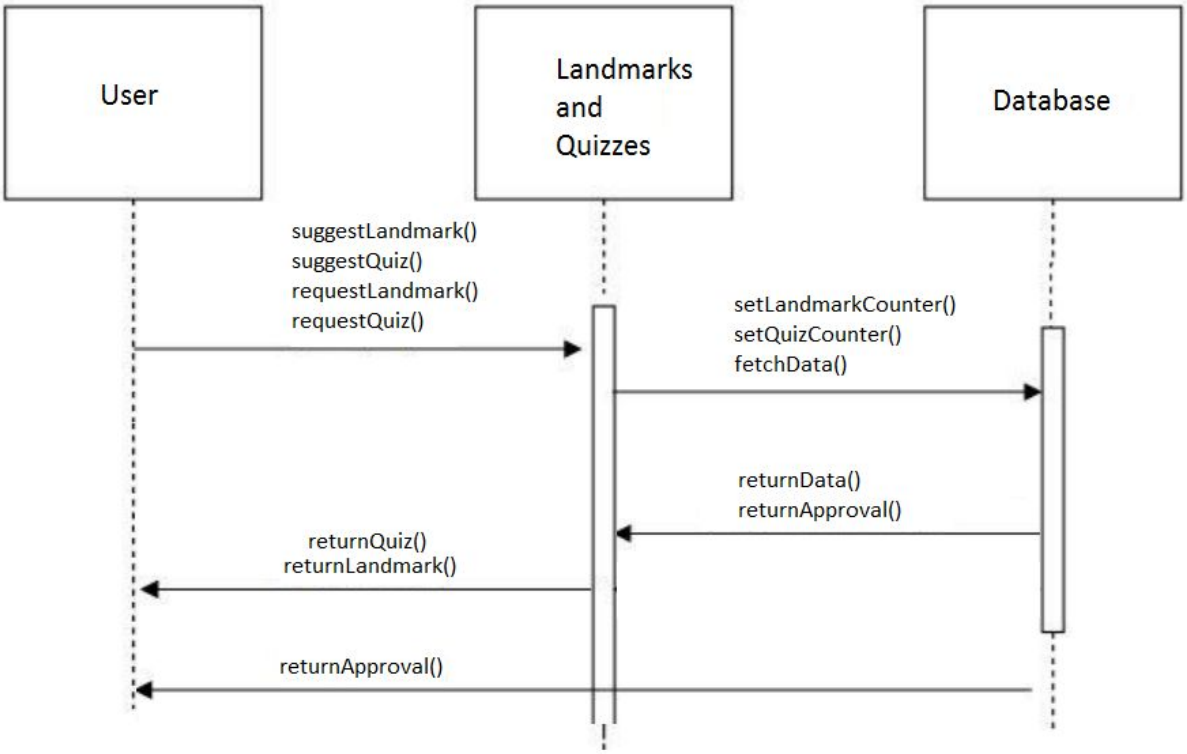
3.b Class Diagrams

The following creates the landmarks and quizzes along with creating the user.

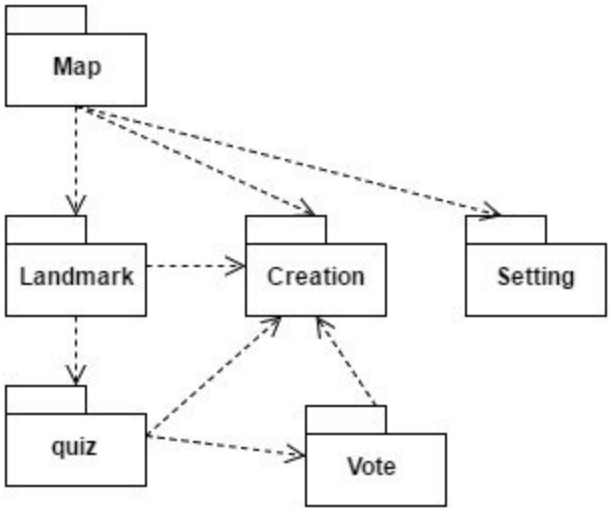


3.c Dynamic Model

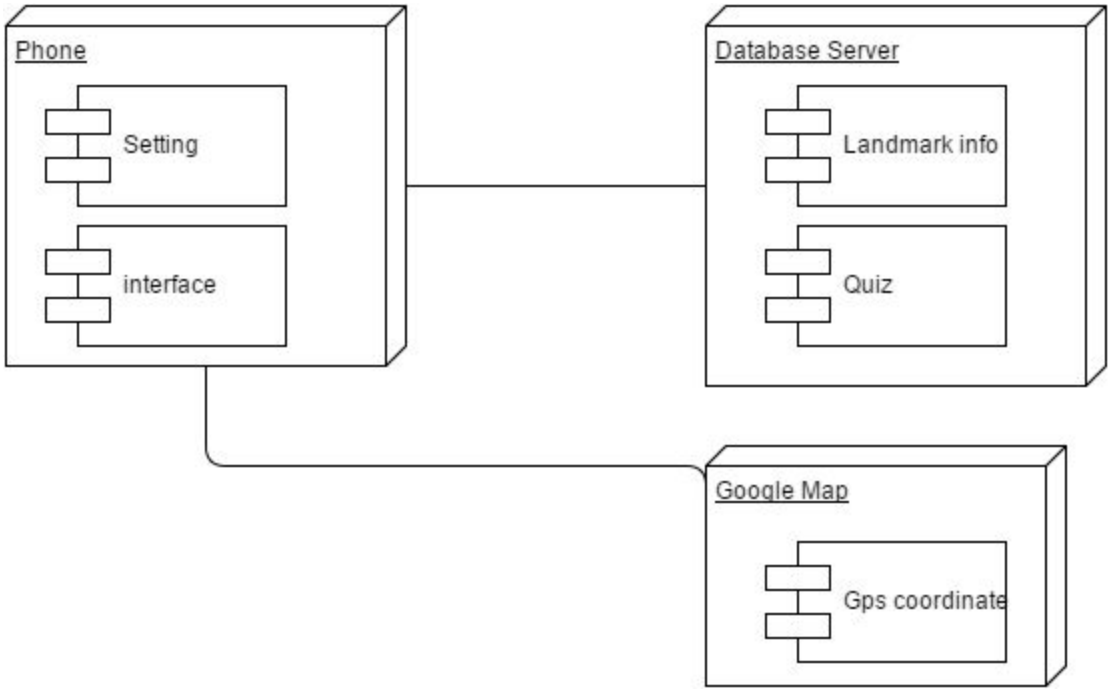
Below is the model for submitting and creating landmarks with the database involved.


User
Landmarks and Quizzes
Database
suggestLandmark()
suggestQuiz()
requestLandmark()
requestQuiz()
setLandmarkCounter()
setQuizCounter()
fetchData()
returnData()
returnApproval()
returnQuiz()
returnLandmark()
returnApproval()

3.d Subsystem Decomposition



3.e Hardware / software mapping



3.f Data Dictionary

Currently Era-set Boulevard does not contain any terminology that should be unfamiliar to our user-base. On top of this, any users that have marginal computer exposure or mobile application usage, should be easily able to understand everything.

3.g Persistent Data management

Any aspects of functionality that must be able to outlive its initial use case and must be able to survive system shut downs or crashes must have a certain level of permanence. These items include:

Profile Information: User profiles and all information linked to the specified user, including level, current experience, and currency, must all be permanent until account deletion, six months of inactivity or intentionally by user or admin.

3.h Access control and security

Access control will be through logging in to profiles for standard users.

For technicians and administrators, direct access to the source code of the mobile application, as well as direct access to servers and the database, through administrative accounts.

3.i Global software control

The system architecture is centralized on the user side, all internal operations dealing with tabase usage is delegated to servers, the users should have no access to any of these operations, only the front end that connects to the internal operations.

3.j Boundary conditions

Boundary conditions are all activities that remain within the scope of Era-set Boulevard, including all activities done through the user interface that is connected to its internal operations.

4 Subsystem services

Map: Allow users to login and load right onto the map.

Landmark: This subsystem contains the landmark, contain the descriptions, and the information of the landmark.

Creation: This subsystem gives the ability to create the landmark and the quiz then submit to voting.

Setting: Allow the user to change their preferences in the program and view the experience points, username and level.

Quiz: This subsystem contains all the quizzes related to the specific landmark .

Vote: Allow user to vote on the landmark or quiz. Sufficient votes shall allow the action be completed, while insufficient votes will place the request on standby.

5 User Interface

The user interface may undergo multiple changes over time to be in the best interest of the userbase and development team, however the following should strictly be included unless disregarded in another time.

The user must be able to see buttons and textboxes to do the following: login/logoff, create/register account, access permissions, settings, view landmark, view quiz, suggest landmark, suggest quiz, vote for landmark/quiz, report quiz, report bug, and/or view product details/about.

The former objects shall be categorized in a way that would seem most logical for the user. Such as logging in shall only include the login button, create/register account, and so on. However the user shall always be able to see the following buttons no matter what screen they're at: report bug, view product details/about, access permissions, and settings.

6 Object Design

6.a Object Design trade-offs

Preliminary design of Era-set Boulevard is meant to be a guideline for the development of the project, not definitive. Adjustments should be made as necessary when developing.

The user interface and functionality is at the developer's discretion, however all implementation front and back end should be shared among the entirety of the team to ensure that the design and functional choices are being made with the user and optimization in mind.

6.b Interface Documentation guidelines

Documentation guidelines should be strictly adhered to. Regardless of lifecycle of the product, the documentation should be sustainable and maintainable, allowing future developers to easily adjust to each iteration of the application.

All classes, packages, and interfaces should be defined with "titlecase."

All methods and class variables should be defined with "camelcase."

All final and global variables should be defined with "uppercase."

6.c Packages

Package classification must be defined by project managers and developers prior to the development of Era-set Boulevard. Defining conventions established prior for interface documentation still apply.

6.d Class Interfaces

Class Interfaces includes:

Landmark

Quiz

Vote

Setting

Gps Interactive

IV Test Plans

1 Features to be tested / not to be tested

Several things need to be tested.

First, make sure the UI is versatile. Some approaches to this would be testing the app out on different devices that greatly vary in screen resolutions. Some things to avoid would be odd screen stretching, aspect ratio anomalies, etc. If any of these appear then the rendering portion of the app will need to be thoroughly debugged to make sure it works fine across all systems. Another thing to look out for is mass clicking of buttons to make sure they don't overlap or do weird things.

The database will need to be thoroughly tested. You can imagine that a lot of people playing at the same time may may prove to be quite taxing on the database and thus needs to be stress tested beforehand so nothing goes wrong on release. Ensuring that the database can handle multiple queries at the same time without stalling would be ideal.

SQL Queries themselves for the database needs to be tested for efficiency so there is the least amount of stalling as possible. Some things this project uses queries for is fetching quiz questions, submitting quiz questions for use-submissions, etc. Best way to test this is with a stress test of several people.

These are the big main 3 that needs to be tested thoroughly, of course there are other things like settings but these three things overshadow all else and efforts should be focused into fine tuning them.

2 Pass/Fail Criteria

Pass Criteria:

- Feature does not fail during stress tests
- No crashes
- Runs efficiently
- Easy to understand at first glance (From a front-end perspective)

Fail Criteria:

- Feature gets overwhelmed/crashes from stress
- Contains memory leaks
- Takes too long to run
- Lack of elegance (Feature doesn't have to be pretty, but can't be created so strangely that users can't intuitively know how to use the feature)

3 Approach

The approach to test our application will rely on people playing the game and send back feedback to fix bugs, improve the interface also any typo or misinformation on the landmark and the quiz will be fixed once the user inform the application.

additional approach to test the application include the white hat hacker on regularly scheduled to test our security to improve our protection of the data in the database.

4 Suspension and resumption

In a perfect environment, the program shall remain running. However due to program optimizations and updates, the program shall go into suspension when a maintenance is required on the servers. Generally, updates do not require a maintenance, only if the update was large enough to affect the application itself. In suspension, the user will not be able to connect to their account, and will be notified that a maintenance is in progress, then brought back to the login screen.

In resumption, the program servers will be back up and users may log back on as usual. Any changes involving the user should be notified.

Additionally, if a website for the program is made, then the website shall report the current service status of the program.

5 Testing materials (hardware / software requirements)

The hardware we will be utilizing for hardware will be low level android phone that can run ice cream sandwich so that an app that can run on that phone and to the modern phone. Any personal computer that run android studios and able to run an emulator.

As we testing on the ice cream sandwich , it will ensure that the application will be able to run on that operation system. some consideration is with future operating system that will be released, some testing must be made to test the compatibility. Also note that as new operating system come out, we should consider stop supporting minimum OS to integrate new feature into the application .

6 Test cases

When performing test cases we use the card below for organization purposes.

Test Case Card: (Name of test case)	(Date Assign)	(Date Due)
•Test Directory:	Directory or folder in which the test case will be held, preferably, include the files affected.	
• Feature to be Tested:	Relevant feature name that will be tested, and provide its description here.	
• Test requirements:	List any requirements that the user must have before performing this test case.	
• Pass/Fail Criteria of - Feature :	List a measurement of what determines a test as a success and what as a fail.	
• Data	Include files, data, or other neccessary information to base this test off of.	
• Test Procedure:	Provide the procedure in performing the task here.	

7 Testing schedule

During the development phase, is expected that all test cases are to be created and formulated, preferably during the coding phase. Tests must be performed in order of development creation. Such scenario is that if the GUI is first created, then tests for the GUI shall be made first before progression. All development phase tests must be

completed before moving onto integration phase, in which integration tests will be performed in a similar manner. Tests shall not be held longer for than 2 weeks, and thus a report must be created if a test is not completed within its time of creation.

A third party such as a white hat hacking organization may be used to do some security stability tests to determine the integrity of the program.

V Project Issues

1 Open Issues

Currently no major issues have arose in the development of this project. Careful focus on the databases and the GPS capabilities of phones that display the locational maps is dire and will mostly likely lead to any open issues in the future.

2 Off-the-Shelf Solutions

2.a Ready-Made Products

Pokemon Go is a finished product that serves as an example solution.

2.b Reusable Components

The largest focus of the app, the gps capability and the landscape creation, is reusable.

2.c Products That Can Be Copied

Pokemon Go stands as the best product example for this genre, and by association, Era-set Boulevard.

3 New Problems

3.a Effects on the Current Environment

Era-set Boulevard stands in a very unique situation, even though Pokemon Go has been ground-breaking for VR mobile games, Era-set Boulevard would be the first of educational VR mobile games to hit the app store. This allows us to not only lead in terms of VR development, but also stand as an example for other educational variants of VR mobile games.

3.b Effects on the Installed Systems

Era-set Boulevard will be made to coexist as much as possible with current systems of today.

3.c Potential User Problems

The largest problems potential users will face, will be lack of internet or slow internet, and outdated phones or version of operating systems for these phones.

3.d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The lack of internet or slow internet will have the largest impact on the product. Everything else should interact without fail should these conditions be met.

3.e Follow-Up Problems

Massive influx of users, or overabundance of logged in users could heavily impact the servers. Optimizing accessing of these servers and their databases, along with a constant vigilance should keep this issue in check.

4 Tasks

4.a Project Planning

The project is meant to be done using agile methods, the method preferable is the SCRUM method, in which sprints can be designed and dedicated for execution by the team.

4.b Planning of the Development Phases

Careful consideration to function vs non-functional requirements should be given when delegating important development tasks amongst members. Time should also be taken to carefully plan out each sprint, with care paid to laying out the foundation and framework of the project first, and interior functionality second.

5 Migration to the New Product

5.a Requirements for Migration to the New Product

In order to properly develop the product as planned and on schedule, a team made up of software developers, software engineers and project managers experienced in software development will be required. This may require hiring additional employees with the desired knowledge and experience. We may also have the need to hire additional employees during the development of the product if it necessary to ensure the proper development of the product or to meet the schedule and given time constraints.

5.b Data That Has to Be Modified or Translated for the New System

As this is an original system, built from the ground up, modification and translation requirements are not needed at this time.

6 Risks

The most serious risks that we face in developing our product include:

- Underestimating the amount of time required to produce our product
- Underestimating cost required to produce product
- Underestimating amount of maintenance to sustain servers
- Underestimating amount of servers needed to accommodate for starting players

7 Costs

The cost of developing the product is largely dependent upon the size of the development team required to produce it and how long they must be employed and paid for in order to meet the demands and requirements for the product discussed in this report. The development team will include software developers, software engineers and project managers experienced in developing large scale software. Each

employee will likely be required to have at a minimum 2 to 3 years of experience in developing software in order to successfully contribute to a software product with this level of sophistication and complexity. The exact cost of each employee will depend on where the product is being developed, as the salary of a software developer is different in each city and location, and their level of knowledge and experience. The expected minimum salary of a developer with the desired level of knowledge and experience is $80,000. The expected salary on the higher end for developers with both more software development experience and experience in management is $150,000. The estimated cost for the budget to produce Era-set Boulevard sits at roughly $5,000,000.

8 Waiting Room

Currently, all desired requirements and features of the product are planned to be included in the first release. However, development of the product may reveal that certain requirements or desired features cannot be implemented according to schedule or the given time constraints. This may require the specified features to be taken out of the first release of the product and instead included in a later release. The schedule and given time constraints, including the possibility of changing them, will have to be considered in determining a final decision.

9 Ideas for Solutions

All solution ideas provided have been implemented into the design of Era-set Boulevard.

10 Project Retrospective

Upon reflection of the development methods that were available to use for this project, the scrum method seems to be the best method. With proper planning and coordination, any errors should be easily avoidable.

VI Glossary

Currently there are no terms that would require a definition for readers of this document to understand the development of Era-set Boulevard.

VII References / Bibliography

No outside references or documents were used in the creation of this report.

VIII Index