Software Requirements Specification

for

Augmented Reality BC Campus Tour App

Version 1.2
15 March 2019

Team: Evgeniya Kokareva, Ana Satkeeva,
Robin Shaw, Bridget Wellen
Supervisor: Dr. Fatma Cemile Serce
CS / Bellevue College

Electronic version Google Drive

Revisions

Version	**Primary Author**	**Description**	**Date Completed**
1.0	All team members	Creating initial draft of SRS as a description for the final product	10/16/2018
1.1	All team members	Minor corrections to each chapter	11/24/2018
1.2	All team members	Minor corrections to each chapter	03/15/2019

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this project is to design a software for AR Campus tour application and defines its functionality. It covers user interfaces, performance considerations, expected operating environment, interfaces with other software and hardware systems, as well as use cases.

1.2 Scope

The AR Campus tour is intended to assist students, professors, and visitors as they navigate the main Bellevue College campus.

The AR Campus tour app is an augmented reality application for Android-powered mobile devices that help out current and future Bellevue College students in recognizing the buildings around campus, learning more information about that particular buildingand finding destinations.

1.3 Definitions, Acronyms, and Abbreviations

- User: Someone who interacts with the mobile phone application
- AR: Augmented Reality
- BC: Bellevue College
- App/ application: AR BC Tour
- RAT: Rational
- DEP: Dependencies
- FR: Functional Requirements
- GPS: Global Positioning System

1.4 References

- ARCore documentation https://developers.google.com/ar/develop/
- Android documentation https://developer.android.com/training/basics/firstapp/
- Augmentatn reality course https://www.udemy.com/develop-augmented-reality-book-ar-business-card-with-unity/learn/v4/content
- Augmented Reality for Developers (Johnathan Linowes, Krystian Babilinski 2017)
- Android Developer Fundamentals Course https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/en/Unit%201/10_c_intro_to_android.html

1.5 Overview

This document contains an overview of what the app will do and what quality standards it will meet. Summary information is provided first with more detailed information in later sections. A complete breakdown is available in the Table of Contents.

2 Overall Description

2.1 Product Perspective

This product will initially be with just a dependence on an Android smartphone with camera but will expand as features are added to work with the Bellevue College website.

2.1.1 User Interfaces

The user will interact with the application via a graphical user interface (GUI). The GUI will consist of a menu structure and popup messages.

2.1.2 Hardware Interfaces

The app requires an extensive use of the device's camera. It will also use the device's screen capture utility. Additionally, the application may utilize the device's GPS location data for tagging image targets. Also, the user will use use the device's "back", "menu", and "home" buttons, will access internet browser and able to make a phone call. All interactions with the hardware will be made through API calls to the Android SDK.

2.1.3 Software Interfaces

Software Interface	Name	Mnemonic	Version	Source
Operating System	Android		>28.0.0	
Database	MySQL Cloud Datastore			
AR Core				
Google Map API				
AWS RDS				

2.1.4 Error Messages

- Building not found
- Unable to identify using image. Please try again.
- That building is not on the BC main campus.
- Cannot start AR session
- Cannot connect to the internet
- Cannot locate user

2.2 Product Functions

Users will be able to use the mobile application to learn information about the various buildings around BC. With AR is implemented, they are able to identify buildings “at a glance” by focusing the building letter in the camera, and the application will provide both information on that building, drawn from the Bellevue College website, and some means of interaction with services housed inside selected building. It will do so with use of Google Maps, the camera, and AR functionalities.

The product will be laid out in a clean, easy to navigate app that takes the user through the launch screen, into some informational menus, and to the AR activity. Help menus will also be available with further explanations and options such as changing the language of the app.

2.3 User Characteristics

Users of this app need to have the ability to use an Android smartphone and simple help menus.

Average Joe is an app user who has a high school diploma and is looking to get into college. He knows how to use his smartphone reasonably well and is looking to this app to help him navigate the campus he is considering attending. He is able to navigate most apps well but has little patience for figuring out complicated user interfaces.

Mama Joe is working hard to help her kids get a good start in life. She finished high school and a little bit of college so knows how important it is to get a good education. She has a smartphone and is able to make calls, read messages, and play some games but is less confident using apps she is new to. She will use an app if it helps her complete her tasks for the day, but gives up easily if the app is too challenging or doesn’t provide the information she wanted. With a younger kid along with her, she is juggling the phone in one hand and doesn’t have time to fidget with lining cameras up just so.

Stephanie French is an international student who has completed high school and some college in France. She has a smartphone and knows how to use it for what she needs. She is able to navigate most apps but can have difficulties with English. Simple interfaces or apps that support her language are easier for her.

2.4 Constraints

Hardware must be an Android device compatible with ARcore and containing a functioning camera. User must grant permissions to access camera and location. App will interface with Google Maps, camera, phone, and location data.

- The programming language shall be Java for the main application.
- The programming language shall be SQL for the cloud targets database. Resource limits.
- The users' device shall have a working camera.
- The users' device shall have a working data plan or wifi connection.
- The users' device shall have sufficient memory storage to install the application.
- The users' device shall have sufficient battery life to run the application.
- The image recognition is possible with a minimum viewing angle of approximately 20 degrees.
- The application does not work in the night view (good lighting needed)
- The service is only offered for buildings on Main BC Campus.

2.5 Assumptions and Dependencies

- Assumes the phone is at least android 8.0 with APK 28 or better in order to run ARcore.
- Assumes the phone has a working camera.
- Assumes the user grants permission for the app to use the camera.
- Assumes the user grants permission to use location data.
- Assumes the functionality of Google Maps.
- Assumes the user is using an android phone.

3 Specific Requirements

This section contains more detailed information regarding the needed functionalities of our application, including interfaces, internal functions, and testing requirements.

3.1 External Interfaces

The app relies on using the camera built into an Android smartphone. The app also interfaces with the Bellevue College website.

3.2 Functional Requirements

Location Module:

Verify user's location

Ex. When the user opens the application, it will prompt the user to either confirm the location or change it to appropriate location.

Recognizing a Building Module:

Find buildings and locations and then turn them into targets to be augmented.

Ex. Once user confirmed the location, application's camera opens and building recognition process starts.

Augmentation Module:
View an augmented targets(letter on the building) through their camera.
Ex. Once user opened camera and application recognized the building, AR layer with menu options open.

Content Builder Module:
Building information should be displayed when camera is aimed at the targeted building. The list of possible operation that can be performed per building should be given.
Ex. Once application recognized the building, menu options appear on AR layer and can be clicked on to expand into more information.

User Interface Module:
Mobile app interface and augmented reality layers should be designed and developed.
Ex. User can go through multiple layers of information once the building is recognized.

Database Module Connect:
Cloud storage to store and retrieve user, building data and augmented files.
Ex. Current view's bitmap (picture) should match one of the bitmaps from database in order to recognize the building for the user.

3.2.1 AR Functionality

3.2.1.1 Functional Requirement 1.1

ID: FR1
TITLE: Camera Permissions
DESC: Ask the user for permission to access their camera.
RAT: Needed for AR functionality.
DEP: None

3.2.1.2 Functional Requirement 1.2

ID: FR2
TITLE: GPS Permissions
DESC: Ask the user for permission to access their location.
RAT: Needed for AR functionality.
DEP: None

3.2.1.3 Functional Requirement 1.3

ID: FR3
TITLE: Recognize Campus Buildings

DESC: Tune the AR functionality to recognize each of the buildings on campus.
RAT: Needed for AR functionality.
DEP: FR1, FR2

3.2.1.4 Functional Requirement 1.4

ID: FR4
TITLE: Overlay Buildings
DESC: Build in the ability to overlay images on the camera via AR
RAT: Needed to hold the information core to our app.
DEP: FR1

3.2.1.5 Functional Requirement 1.5

ID: FR5
TITLE: Pair Buildings to Overlays
DESC: Overlay the correct layer to the correct building.
RAT: This holds the information key to our app.
DEP: FR3, FR4

3.2.1.6 Functional Requirement 1.6

ID: FR6
TITLE: Interact with Overlays
DESC: Build in interactive features to the overlays.
RAT: Extends the features of our app.
DEP: FR3

3.2.1.7 Functional Requirement 1.7

ID: FR13
TITLE: Phone Permissions
DESC: Ask the user for permission to make phone calls.
RAT: Extends the features of our app.
DEP: None

3.2.2 UI Interfaces

3.2.2.1 Functional Requirement 2.1

ID: FR7
TITLE: Help/Options Menus
DESC: Program and allow the user access to a help/options menu.

RAT: To hold several useful options.
DEP: None

3.2.2.2 Functional Requirement 2.2

ID: FR8
TITLE: Custom Language
DESC: Allow the user to alter the language the app displays in.
RAT: To service our ESL clients.
DEP: FR7

3.2.2.3 Functional Requirement 2.3

ID: FR9
TITLE: Draw from the Internet
DESC: Allow the app to draw information listed on the internet.
RAT: To provide the information needed for some features.
DEP: None

3.2.2.4 Functional Requirement 2.4

ID: FR10
TITLE: Confirm Location Button
DESC: On the opening screen, takes selected location and sends it to the next activity.
RAT: Needed for location information.
DEP: None

3.2.2.5 Functional Requirement 2.5

ID: FR14
TITLE: Display to Textview
DESC: Create a method that allows a textview to display the given information
RAT: Useful for UI
DEP: None

3.2.2.5 Functional Requirement 2.6

ID: FR15
TITLE: Display Recyclerview
DESC: Create an easy to implement recyclerview
RAT: Useful for UI
DEP: None

3.2.3 Database Functionalities

3.2.3.1 Functional Requirement 3.1

ID: FR11
TITLE: Confirm Database Connection
DESC: Ensure the database is properly connected and working
RAT: Needed to draw image/location data from the database for AR functionality
DEP: None

3.2.3.1 Functional Requirement 3.2

ID: FR12
TITLE: Withdraw Database Information
DESC: Withdraw picture and location information from the database
RAT: Needed for AR functionality
DEP: FR11

3.3 Performance Requirements

Camera lag should be no more than 0.2 seconds. The app will support 10,000 simultaneous users. The app should be available at least 95% of the time between the hours of 6:00am and 11:00pm with all maintenance and scheduled down time occurring during gate closure hours. After clicking any button, user shall receive a response within 1 second.

3.4 Logical Database Requirements

The building info, user info, and augmentation will be stored in a database. We will use cloud data storage services for this purpose.

3.5 Design Constraints

The AR needs to be lightweight enough that it can run with minimal delay on a standard Android smartphone. The app should be as small a size as possible because storage on phones is important. Information provided by the app may be limited to what is available on the internet. The app will not have access to any student's private data.

3.6 Software System Attributes

3.6.1 Adaptability

- Any smartphone running an Android system of at least version 28.0.0 shall be able to run the application
- Any user with smartphone or tablet experience shall be able to comfortably use the application
- Learning curve for this app shall be very low as the user interface is easy to navigate

3.6.2 Reliability

- The app will not crash or hang except as the result of operating system error

3.6.3 Availability

- The app will be available at least 95% of the time between the hours of 6:00am and 11:00pm
- Download and install application shall be available through Google App store

3.6.4 Security

- There are no specific requirements

3.6.5 Maintainability

- The app will be as self contained as possible to allow for ongoing maintenance.

3.6.6 Portability

- Not available on any platform other than Android