FIT History Tours - Software Requirements Specification

10/3/2022

Table of Contents

1. Introduction

1.1. Purpose

This document contains the Software Requirements Specification for the FIT History Tours Software Development Project. It explains this project and its requirements, functional or otherwise, to the fullest extent understood by its authors. The subsequent documentation has been written with an intended audience consisting of:

- Dr. Philip Chan
- Dr. Ryan Stansifer
- Dr. Fitzroy Nembhard
- Peers and other students of Florida Institute of Technology

1.2. Scope

These requirements were elicited for the development of the "FIT History Tours" mobile application.

This software will provide a client application on iOS, Android, and in-browser, that allows users to embark on geolocation-guided tours within the Florida Institute of Technology campus bounds, engaging with historical content relevant to the landmarks visited during the tour. Also, the software will allow users not present on the campus to navigate a virtual tour that contains the same historical content as the guided tour. This software will *not* replace the current admissions tour process nor serve the purpose of introducing prospective students to the campus prior to enrollment.

The working scope of this project includes historical education, automated tour guidance, and social enrichment. The high-level goal for the software is to provide users with an interactive guided tour experience through the campus of Florida Institute of Technology such that users may enhance their knowledge of the university's unique history and background on an at-will basis. Users will benefit from the software through informational exposition, flexibility to tour campus when they please (within normal working hours), and client-side activities that provide further engagement with the educational content.

1.3. Definitions, Acronyms, and Abbreviations

Term or Acronym	Definition
FIT, Florida Tech	Abbreviations for *Florida Institute of Technology*

Tours app	An alias to represent FIT History Tours app in fewer words
GCP	Google Cloud Compute
AWS	Amazon Web Services
DO	DigitalOcean
IP	Internet Protocol
CDN	Content Delivery Network
SSL	Secure Sockets Layer
TLS	Transport Layer Security

1.4. References

The following documents and articles were referenced for the creation of this document, in addition to client and advisor correspondence:

1.4.1. **830-1998 - IEEE Recommended Practice for Software Requirements Specifications**: Link
1.4.2. **Apple App Store Review Guidelines**: Link
1.4.3. **Uptime Requirements**: Link
1.4.4. **Google Play Store Developer Policy**: Link

1.5. Document Overview

The remainder of this document contains the details and requirements of the Tours app software. Section 2 concerns the general description of the software and its functions, with dedicated subsections for each aspect. Section 3 formally details the explicit requirements that the software must meet for acceptance. Section 4 contains any additional appendices and relevant reference models that may be referred to in the preceding sections. Any underlined text has a hyperlink embedded to bring you to the corresponding resource.

2. Software Description

2.1. Product Perspective

Florida Tech History Tours will be an application independent from all university run applications and serve the sole purpose of providing students, staff, and other users with the ability to tour campus with historical facts and games. Florida Tech provides other applications and methods for providing informational tours on campus about student life, building functions, and more but this application will be focused on historical information. This product is self contained with the external information only being the historical information collected, cataloged, and monitored by the Florida Institue of Technology. That is outside the scope of this application, but that information will be inputted into the application for usage in timelines and other features of the application.

This application will be platform agnostic with all devices with a web-browser and location services being supported. The main system interfaces will be with the devices the user uses to open and navigate the application. The app will function with either the touchscreen or mouse pointer of various device types (mobile phone, touchscreen laptops, laptops, or PCs). The user interface will have a navigation menu for either mode (virtual or in-person); otherwise, there will be more user interfaces described in the design document. The application will interface with the hardware through the web-browser to access various data such as the location, mouse movements, and keyboard input. The application will be web-hosted with IP connectivity to all devices wishing to take a FIT History Tour. The application will be publicly accessible with a DNS name so the user can connect easily.

2.2. Product Functions

The Tours app will achieve the following general functional objectives:

2.2.1. **Guided Tours** - The Tours app will make use of a geolocation API to assess user location and display on the user device a road map of the Florida Tech campus, centered on the user’s last recorded location. During a tour, the user interface will dynamically guide the user to the next selected point of interest based on collected and existing location information. At each point of interest, the software will present the user with relevant historical information about the location.

2.2.2. **Virtual Tours** - The Tours app will contain a road map of the Florida Tech campus with notation to convey the set of historical points of interest available for user interaction. Upon selection of a point of interest, the software will display relevant historical information to the user about that location.

2.2.3. **Historical Trivia & Games** - The Tours app will allow users to play and interact with a scavenger hunt and trivia to continue their engagement with the application. These features will allow users to both explore further into campus and the places they are visiting while also keeping it engaging for people who use the app multiple times. Trivia will have multiple levels of difficulty while the scavenger hunt will be tailored to the area of the university you are currently in.

2.3. User Characteristics

Users of the application should be able to do various activities based on their physical location and preferences. The application will support and cater to two distinct types of users, users that are physically present on the Florida Tech campus, and users that are not physically present on the Florida Tech campus. Features designed for users that are physically on campus will not be available to those who are not physically on campus, however all features that are designed for users that are not physically on campus will be available to users that are physically on campus. The application will encourage users that are present on campus to participate in the geolocation-guided activities, but will not stop users from also accessing the virtual tour or any other non-geolocation-guided activities.

Users that are physically present on the Florida Tech campus should be able to:

- Participate in geolocation-guided tours
- Participate in geolocation-guided scavenger hunts
- Participate in either geolocation-guided or non-geolocation-guided trivia games
- Participate in a virtual tour
- Access an “active timeline” that updates as the user physically or virtually explores campus

Users that are not physically present on the Florida Tech campus should be able to:

- Participate in a virtual tour
- Participate in non-geolocation-guided trivia games
- Access an “active timeline” that updates as the user virtually explores the campus’ history

Whether the user is physically present on the Florida Tech campus, the user should be able to get a good understanding of the University’s history no matter which activities are available to them. Some features are heavily geolocation dependent and thus, those specific features are available exclusively for users who are physically able to access campus. Users should have a choice in which activities they wish to participate in and instead of forcing users to participate in specific activities tailored to which category they belong to, the application should adapt to allow every user to get as much information as possible out of the app, no matter their means.

2.4. Constraints

The options afforded to developers for this project will be limited by the following constraints:

2.4.1. **Regulatory Policies**

These policies restrict the interference of the software with other devices and impact minimum acceptance requirements for production and release of a mobile application. Requirements defined within these documents will not be repeated in section 3. The regulations followed include:

2.4.1.1. Apple App Store Review Guidelines [1.4.1.]
2.4.1.2. Google Play Store Developer Guidelines [1.4.4.]

2.4.2. **Hardware Limitations**
The devices this application will be tested on and utilized on must have the capability to connect to the internet with WiFi, cellular data, or other modes of connection. The devices must also have the ability to run a web browser. There are no specific device performance requirements if these two constraints are met.

2.4.3. **Language Limitations**
The language this project is constrained by the cross platform aspect. The chosen high level language should be able to run on any device (mobile or desktop).

2.4.4. **Reliability Limitations**
The application does not need to maintain a 99.99% uptime but will need to provide over 90% uptime throughout the year. This constraint is placed to ensure that the application is usable by the general public for a majority of the year. This reliability needs to be mainly in application and database stability to ensure that the application does not crash due to issues in the codebase. The requirement for the hosting provider is 99.99% uptime, and hence this constraint is placed solely on the application and database built by the group.

2.5. Dependencies

2.5.1. **Hosting Platforms -** This application will be dependent upon the selected hosting platform and their uptime. As all hosting services (AWS, GCP, DO, and self-hosting) are prone to downtime, the dependency upon one is major as with the hosting service going down, our application will go down as well. All of the previously mentioned hosting providers have an uptime of 99.9% advertised across years of monitoring. This dependency isn't something major for our group as our application isn't going to be mission critical for users and businesses; rather something to be noted on all web hosted applications.

2.5.2. **Information and Imagery -** This application is also dependent on information and imagery provided by Florida Tech and other online sources. This historical information will be of varying levels of quality and quantity. Images, news articles, and other information collected can be subject to copyright and other legal protections so proper approval will need to be obtained for all information hosted in the application and any historical databases built for the application.

2.5.3. **Database -** SQL, MySQL, DynamoDB, or any other database software that is used will be another dependency as that service will provide updates and patches for our application backend database. This database will need to have proper documentation and support for any issues the project may encounter.

2.5.4. **External NPM Packages -** Node Packages for different aspects of the JavaScript portion of the project will be used and as these are kept up to date by external developers each will be a dependency. As the project grows and we add new packages to the project bundle, each will need to be checked that they are maintained and don't have any bugs in the current version that would degrade the performance or usability of the application.

2.5.5. **IP Connectivity -** This application will be loaded from the web requiring the presence of IP connectivity to gather items from the server where the application is hosted. The application may keep some information locally on the device but the main portion of the historical information and guided tours will be done on a server externally from the device requiring connectivity to get information from the server.

2.6. Future Requirements

These requirements are future ideas proposed by the client that are outside the scope of the proposed project but may be added at a later point to improve the quality of the application.

2.6.1. AI Model to Reccomend Next Location

While a user is on a tour, an AI can choose the next location the user might want to visit based off some input from the user. The idea of where they had been, a survey to start, or other forms of data collection could be input to a model to allow the model to provide the user with tidbits of history they might find interesting.

2.6.2. 3D Visualization of Buildings in Augmented Reality

The application can drop 3D visualziations of previous buildings into an Augmented Reality setting during a tour. The previous buildings or images converted into 3D visualziations could be placed down on the ground by using the cameras of the device a person is using.

3. Specific Requirements

3.1. External Interfaces

3.1.1 **User Interfaces** - The user will be able to use the user interface to interact with the application, accessing virtual and guided tours, freely roam landmarks, and access the database of historical data.

3.1.2 **Hardware Interfaces** - The software would primarily interact with the application through a touchscreen on a mobile device. The software will interface with the GPS modules in the device to acquire location information.

3.1.3 **Software Interfaces** - The application will interface with the geolocation services on iOS and Android, along with Safari, Firefox, and Chrome browsers. The minimum browser requirements to run the application are as follows:

- Safari 10.1+
- Google Chrome 60+
- Firefox ESR+

3.2. Functions

3.2.1. Persistent Requirements

3.2.1.1. The software shall display a road map of the Florida Tech campus.

3.2.1.2. The software shall display waypoints at the primary entrance to points of interest 75% of the time.

3.2.1.3. The software shall access the user's geolocation data N times per M minutes and utilize accelerometer values to represent the user's position on the display map.

3.2.1.4. The software shall enable the user to access a settings menu.

3.2.1.4.1. The settings menu shall contain an option to enable or disable usage of geolocation services.

3.2.1.4.2. The settings menu shall contain an option to enable or disable the scrolling timeline [3.2.2.3.].

3.2.1.5. The software shall display an option to exit tours currently in progress.

3.2.2. Mode 1: First Load

3.2.2.1. The software shall prompt the user to allow usage of geolocation info.

3.2.2.1.1. If the user allows access:

- Continue from 3.2.2.2.

3.2.2.1.2. If the user rejects access:

- Continue from 3.2.2.2.2.

3.2.2.2. The software shall display a pop-up dialog recommending the user to initiate a tour activity.

3.2.2.2.1. If the user is on Florida Tech grounds:

- The dialog will prompt the user to begin a guided tour.
- If selected, the software shall enter Mode 2, Guided Tour.

3.2.2.2.2. If the user is not within the bounds of Florida Tech's campus:

- The dialog will prompt the user to begin a virtual tour.
- If selected, the software shall enter Mode 4, Virtual Tour.

3.2.2.2.3. If the user selects "No" or clicks away from the prompt:

- The dialog will close and the software will enter Mode 3, Free Roam.

3.2.2.3. The software should display a scrollable timeline that spans the years 1958 to the present year.

3.2.3. Mode 2: Guided Walking Tour In Progress

3.2.3.1. The software shall direct the user along paths on Florida Tech campus to navigate towards the currently targeted point of interest.

3.2.3.2. The software shall determine user proximity to the currently targeted point of interest.

3.2.3.2.1. If the user is less than 10 meters away from the point of interest:

- The software shall display historical information about the nearby point of interest.
- The software shall allow the user to navigate through additional historical information related to the point of interest.

3.2.3.3. The software shall mark the currently targeted point of interest as "visited," and save this designation locally.

3.2.3.4. The software shall provide greater than or equal to 3 and less than or equal to 5 options for the user to select as the next targeted point of interest.

3.2.3.4.1. The software shall display options that have not been saved as visited.

3.2.3.4.2. The software shall prioritize the options by distance from the currently targeted point of interest.

3.2.3.5. The software shall allow the user to defer selection of the next targeted point of interest to a predetermined tour order.

3.2.4. Mode 3: Free Roam

3.2.4.1. The software shall display the standard road map of Florida Tech with all points of interest available for selection.

3.2.4.2. The software shall provide historical information for one selected point of interest.

3.2.4.3. The software shall provide an option for directions to the currently selected point of interest.

3.2.4.4. The software shall display an option for the user to begin a tour.

3.2.5. Mode 4: Virtual Guided Tour In Progress

3.2.5.1. The software shall begin a virtual tour by targeting the Clemente Center.

3.2.5.2. The software shall display historical information about the currently targeted point of interest.

3.2.5.2.1. The software shall allow the user to navigate through additional historical information related to the point of interest.

3.2.5.3. The software shall allow the user to navigate to the next point of interest in a predetermined tour order.

3.2.6. Mode 5: Scavenger Hunt In Progress

3.2.6.1. The software shall begin a scavenger hunt with the point of interest closest to the user upon initialization.

3.2.6.2. The software shall determine user proximity to the currently selected point of interest.

3.2.6.2.1. If the user is less than or equal to 15 meters away from the point of interest:

- The software shall prompt the user to complete a number of goals greater than zero and less than or equal to five in the area less than or equal to 15 meters away from the point of interest that relates to historical information contained in the software database.

3.2.6.2.2. If the user is greater than 15 meters away from the point of interest:

- The software shall provide a hint to guide the user towards the point of interest.
- The software shall provide an option to be directed to the point of interest.

3.3. Performance

3.3.1. **Capacity**

3.3.1.1. The webserver server should be able to provide greater than or equal to 20 simultaneous tours at one time.

3.3.1.2. The application server should be able to handle 50 viewers browsing and interacting with the timeline while not on a tour.

3.3.1.3. The link to each device should be able to stream a 320kbps audio file without any audio tearing or crackling. This represents a minimum bandwidth requirement of 8Mbps for the server.

3.3.2. **Static and Dynamic Performance**

3.3.2.1. The webserver should be able to respond to the devices current location within one and a half seconds.
3.3.2.2. The application loadtime should be less than three seconds to being able to input something into the start screen.
3.3.2.3. The application should respond to on screen user input with one second. This can be either with a touchscreen or a mouse/keyboard.
3.3.2.4. The application can continue to track location in the background while user is moving without taking up more than 20% of the browsers resources.
3.3.2.5. Scrolling through the timeline should return new facts within one second to fill any boxes that may appear on the screen.
3.3.2.6. Images should appear on the users application within one second of being loaded onto the screen.

3.4. Database Requirements

3.4.1. **Database Overview -** FIT History Tours will be able to import the database in any of the following database servers. While the minimum version will work, it is recommended to use the most recent version of any database for security and functionality updates. These databases support a wide array of deployment types and schemas, but each can be used to load the database into.

3.4.2. **MySQL -** The minimum version of MySQL is MySQL 5.7.20. The most recent version is recommended.

3.4.3. **MariaDB -** The minimum version of MariaDB is MariaDB 10.6. The most recent version is recommended.

3.4.4. **MongoDB -** The minimum version of MongoDB is MongoDB 5.0.12. The most recent version is recommended.

3.5. Design Constraints

As previously mentioned in earlier sections, the application will be focused on geo-location guided activities. However, activities that don’t depend on a user’s geo-location will also be available. Since some users won’t be physically present on campus when accessing the application, the application must be designed to accommodate both types of users. For example, the application’s main feature, the geolocation-guided tour, will not be possible to any users not physically present on campus, a virtual tour will take its place which would give a good understanding of the campus’ history and allow users to be able to participate in trivia games. While the application will be heavily centered on the geolocation-guided activities, it would be alienating to only offer such activities.

Another design constraint to consider is the availability of material and information. All information within the tours and activities needs to be accurate and verifiable by a credible source. The length of the tours and activities is limited by how much is verified and available.

3.6. Software Attributes

There are a number of attributes of software that can serve as requirements. It is important that required attributes be specified so that their achievement can be objectively verified. Subclauses 5.3.6.1 through 5.3.6.5 provide a partial list of examples.

3.6.1. **Reliability**

The application will need to maintain usability during peak times of the day when there may be heavy traffic. The application is not mission critical so the reliability will not be 99.99% but will generally be above 90% (less than 36 days a year). Ideally, 97% can be achieved by ensuring that the hosting provider has the functionality to restart the webserver if it was to go down during normal operation. A load balancer will also be implemented during deployment to keep the reliability higher since the application will dynamically scale to allow more users.

3.6.2. **Availability**

The application should be reachable around the United States of America during any conditions. The application will be placed behind a CDN allowing for high speed availability around the United States. The application will be hosted by a hosting provider that will handle the deployment of the web application across datacenters and DNS nodes across the United States to ensure the application is available across the region.

3.6.3. **Security**

The application will not store any user data between visits including passwords, names, or other personal information. Hence, the main security for the application will be ensuring the internal systems don’t get compromised to allow others to take over or run malicious code in the application or on the hosting virtual machine. All traffic with the website will be made over secured SSL/TLS to ensure all traffic is secured while it travels across the network to the users device. All internal systems will remain on the same virtual machine to prevent the systems from being broken off from one another. While the database, webserver, and other items may be deployed to different servers to provide more security, this isn’t possible in the current budget constraints. The database will remain static unless updated by a group member but a trigger will alert the group Discord of any updates to the database. The webserver can only be updated by GitHub commits to the main branch after they are checked by another member of the group. The deployed code will be checked against the main branch of the GitHub codebase to ensure that there hasn’t been any maliciously added code.

3.6.4. **Maintainability**

The software will be easily maintainable by the group. The codebase updates will be managed through GitHub with automatic deployments to our production server. This will allow us to make changes to the code at anytime anywhere without need physical or digital access to the hosting server itself. As for the database, after the initial load of data it won't need to be touched unless more data is to be added to it. If data is to be added, the specifications for addition will be explained in the user and administration manual upon completion of the project.

3.6.5 **Portability**

The code will be written almost entirely in JavaScript, allowing it to be run on any platform that supports the node-js compiler. The databases chosen are supported on a large number of operating systems including MacOS, Windows, and Debian Linux. The ReactJS framework can be generated into static website pages that can be run off of a webserver on any operating system. There is zero host dependent code running off the system.