

FIT History Tours Design Document
10/1/2022

Table of Contents

1. Overview	4
1.1 Scope	4
1.2 Purpose	4
1.3 Intended audience	4
1.4 Conformance	4
2. Definitions and Acronyms	5
3. Conceptual model for software design descriptions	6
3.1 Software design in context	6
3.2 Software design descriptions within the life cycle	6
4. Design description information content	7
4.1 Introduction	7
4.2 SDD identification	7
4.3 Design stakeholders and their concerns	7
4.4 Design views	7
4.5 Design viewpoints	8
4.6 Design elements	9
4.7 Design overlays	9
4.8 Design rationale	9
4.9 Design languages	9
5. Design viewpoints	11
6. Human Interface Design	12
6.1 Overview of User Interface	12
6.1.1 Free Roam	12
6.1.2 Guided Tour	12
6.1.3 Trivia	12
6.2 Screen Images	13
6.2.1 Free Roam	13
6.2.2 Virtual Tour	14
6.2.3 Guided Tour	15
6.2.4 Trivia	16
6.2.5 Navigation Menu	17
6.2.6 Settings Menu	18

6.2.7 Point of Interest Example	19
6.2.8 Scavenger Hunt Example6.2.8 PC Virtual Tour	20
6.3 Screen Objects and Actions	22
6.3.1 Map	22
6.3.1.1 Points of Interest	22
6.3.2 Timeline	22
6.3.3 Side Menu	22
6.3.3.1 Hamburger Button	22

1. Overview

1.1 Scope

This document is designed to give a detailed and technical description of the FIT History Tours application. Details relating to the design, functions and deliverables of the application will be given, however, details relating to the applications general use will not be given in this document.

1.2 Purpose

The history of the Florida Institute of Technology is one that is well documented but not known widely by the general public. The FIT history tours application intends to inform the public of FIT's history through the medium of interactive tours, games, and a rudimentary timeline. This document describes and showcases how this software will accomplish its goals and requirements.

1.3 Intended audience

The intended audience of this document consists of:

- Dr. Philip Chan
- Dr. Ryan Stansifer
- Dr. Fitzroy Nembhard
- Anyone who intends or needs to understand the FIT History Tours application implementation.

The reader needs to have some knowledge of software engineering.

1.4 Conformance

This document conforms to the standards set by IEEE std 1016-2009.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5167255&tag=1>

2. Definitions and Acronyms

Term or Acronym	Definition
FIT, Florida Tech	Abbreviations for <i>Florida Institute of Technology</i>
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
UI	User Interface

3. Conceptual model for software design descriptions

These Software Design Descriptions (SDDs) being presented in this document attempt to encapsulate the complexities of decisions made throughout the design process, including basic explanations of complex concepts, how they are prepared, stakeholders who use them, and how they are utilized.

3.1 Software design in context

The Tours app is an interactive resource for historical data regarding Florida Tech, including the campus and surrounding events that shaped the school as it grew. The application provides this information by showing the user their current location on campus, pointing out points of interest to the user, guiding them on a tour, giving them access to the database of information and giving them a virtual tour when they are not on campus.

3.2 Software design descriptions within the life cycle

The SDDs are made to be used within the life cycle of development on FIT Tours App, including but not limited to the duration of the 2022-2023 Fall and Spring semesters. The life cycle includes standards for outcomes as defined by IEEE 12207-2008, which state providing the deliverables of the project. For this project, the FIT Tours App being deployed and ready to be used by current students at Florida Tech with all of its described features is the main deliverable.

4. Design description information content

This section follows the standards of IEEE 1016-2009, providing context about how the SDDs in this document are written.

4.1 Introduction

The Software Design Descriptions (SDDs) in this document contain:

- Identification of the SDD
- Identified design stakeholders
- Identified design concerns
- Selected design viewpoints, each with type of definitions of its allowed design elements and design languages
- Design views
- Design overlays
- Design rationale

4.2 SDD identification

An SDD shall include the following descriptive information:

- Date of issue and status
- Scope
- Issuing organization
- Authorship
- References
- Context
- One or more design languages for each design viewpoint used
- Body
- Summary
- Glossary
- Change history

4.3 Design stakeholders and their concerns

An SDD shall identify the design stakeholders for the design subject.

An SDD shall identify the design concerns of each identified design stakeholder.

An SDD shall address each identified design concern.

4.4 Design views

An SDD shall be organized into one or more design views.

Each design view in an SDD shall conform to its governing design viewpoint.

The purpose of a design view is to address design concerns pertaining to the design subject, to allow a design stakeholder to focus on design details from a specific perspective and effectively address relevant requirements.

Each design view shall address the design concerns specified by its governing design viewpoint.

An SDD is *complete* when each identified design concern is the topic of at least one design view; all design attributes refined from each design concern by some viewpoint have been specified for all of the design entities and relationships in its associated view; and all design constraints have been applied.

An SDD is consistent if there are no known conflicts between the design elements of its design views.

4.5 Design viewpoints

For each design view in an SDD, there shall be exactly one design viewpoint governing it.

Each design viewpoint shall be specified by:

- Viewpoint name;
- Design concerns that are the topics of the viewpoint;
- Design elements, defined by that viewpoint, specifically the types of design entities, attributes, relationships, and constraints introduced by that viewpoint or used by that viewpoint (which may have been defined elsewhere). These elements may be realized by one or more design languages;
- Analytical methods or other operations to be used in constructing a design view based upon the viewpoint, and criteria for interpreting and evaluating the design; and
- Viewpoint source (e.g., authorship or citation), when applicable.

In addition, a design viewpoint specification may provide the following information on using the viewpoint:

- Formal or informal consistency and completeness tests to be applied to the view;
- Evaluation or analysis techniques to be applied to a view; and

- Heuristics, patterns, or other guidelines to assist in construction or synthesis of a view.

An SDD shall include a rationale for the selection of each selected viewpoint.

Each design concern identified in an SDD (4.3) shall be framed by at least one design viewpoint selected for use. A design concern may be the focus of more than one viewpoint in an SDD.

4.6 Design elements

A design element is any item occurring in a design view. A design element may be any of the following subcases: design entity, design relationship, design attribute, or design constraint.

4.7 Design overlays

A design overlay is used for presenting additional information with respect to an already-defined design view.

Each design overlay shall be uniquely named and marked as an overlay.

Each design overlay shall be clearly associated with a single viewpoint.

4.8 Design rationale

Design rationale captures the reasoning of the designer that led to the system as designed and the justifications of those decisions.

Design rationale may take the form of commentary, made throughout the decision process and associated with collections of design elements. Design rationale may include, but is not limited to: design issues raised and addressed in response to design concerns; design options considered; trade-offs evaluated; decisions made; criteria used to guide design decisions; and arguments and justifications made to reach decisions.

4.9 Design languages

Design languages are selected as a part of design viewpoint specification (4.5).

A design language may be selected for a design viewpoint only if it supports all design elements defined by that viewpoint.

Design languages shall be selected that have:

- A well-defined syntax and semantics; and
- The status of an available standard or equivalent defining document.

Only standardized and well-established (i.e., previously defined and conveniently available) design languages shall be used in an SDD. In the case of a newly invented design language, the language definition must be provided as a part of the viewpoint declaration.

5. Design viewpoints

Referring to the IEEE 1016-2009 document's section 5 will show the different viewpoints that are to be used in any SDDs in this document. A table of these viewpoints and their rationale from that document:

Table 1—Summary of design viewpoints

Design viewpoint	Design concerns	Example design languages
Context (5.2)	Systems services and users	IDEF0, UML use case diagram, Structured Analysis context diagram
Composition (5.3) Can be refined into new viewpoints, such as: functional (logical) decomposition, and run-time (physical) decomposition.	Composition and modular assembly of systems in terms of subsystems and (pluggable) components, buy vs. build, reuse of components	Logical: UML package diagram, UML component diagram, Architecture Description Languages, IDEF0, Structure chart, HIPO Physical: UML deployment diagram
Logical (5.4)	Static structure (classes, interfaces, and their relationships) Reuse of types and implementations (classes, data types)	UML class diagram, UML object diagram
Dependency (5.5)	Interconnection, sharing, and parameterization	UML package diagram and component diagram
Information (5.6) with data distribution overlay and physical volumetric overlay	Persistent information	IDEF1X, entity-relation diagram, UML class diagram
Patterns (5.7)	Reuse of patterns and available Framework template	UML composite structure diagram
Interface (5.8)	Service definition, service access	Interface definition languages (IDL), UML component diagram
Structure (5.9)	Internal constituents and organization of design subjects, components and classes	UML structure diagram, class diagram
Interaction (5.10)	Object communication, messaging	UML sequence diagram, UML communication diagram
State dynamics (5.11)	Dynamic state transformation	UML state machine diagram, statechart (Harel's), state transition table (matrix), automata, Petri net
Algorithm (5.12)	Procedural logic	Decision table, Warnier diagram, JSP, PDL
Resources (5.13) May be refined into resource based viewpoints with possible overlays	Resource utilization	UML Real-time Profile, UML class diagram, UML Object Constraint Language (OCL)

6. Human Interface Design

6.1 Overview of User Interface

The application's UI offers several different modes of interaction for the user to choose between as they use it. At all times the user has access to the settings and always has the ability to enter any mode they would like to, including Free Roam, Guided Tour, and Trivia.

6.1.1 Free Roam

This mode is the base state of the app, where the user is able to move freely and interact with any nearby points of interest on the map while having access to a timeline of information on that particular point of interest.

6.1.2 Guided Tour

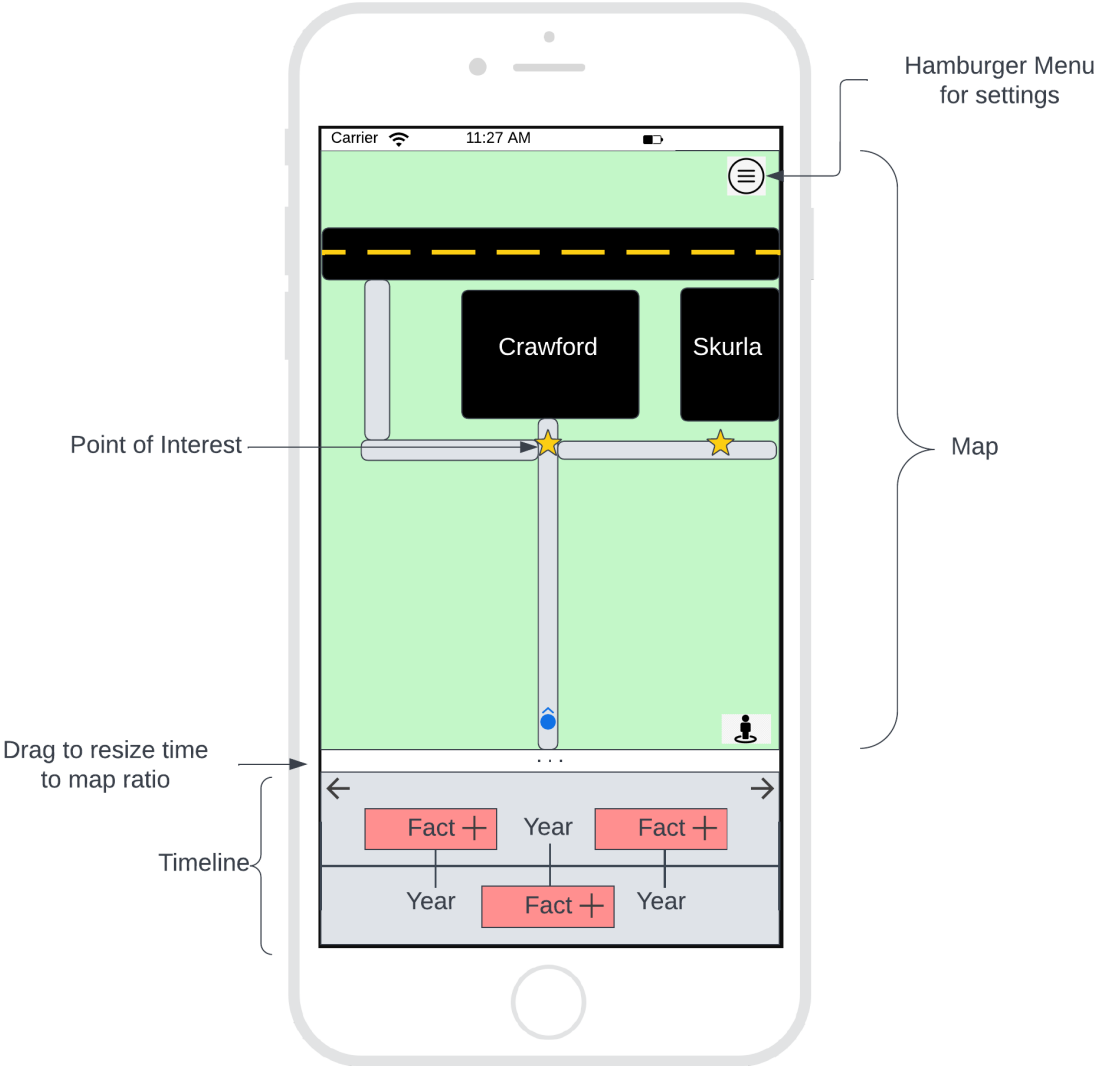
In Guided Tour, the FIT Tours app will have the user go towards a point of interest on the map, opening information about it as the user moves closer. This mode has the user interacting through both the app and in person as they learn about the point of interest.

6.1.3 Trivia

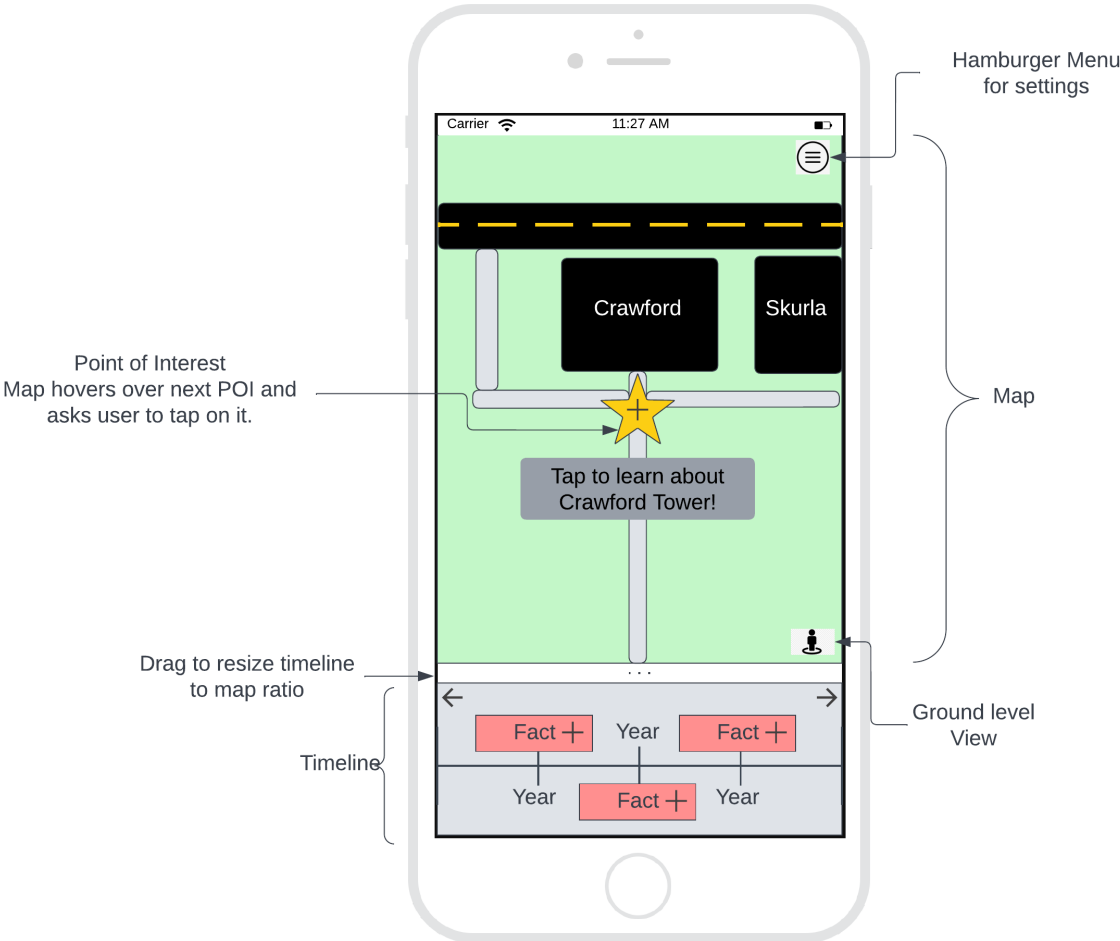
Trivia mode prompts the user to answer a series of questions about the historical information, keeping track of their score as they answer.

6.2 Screen Images

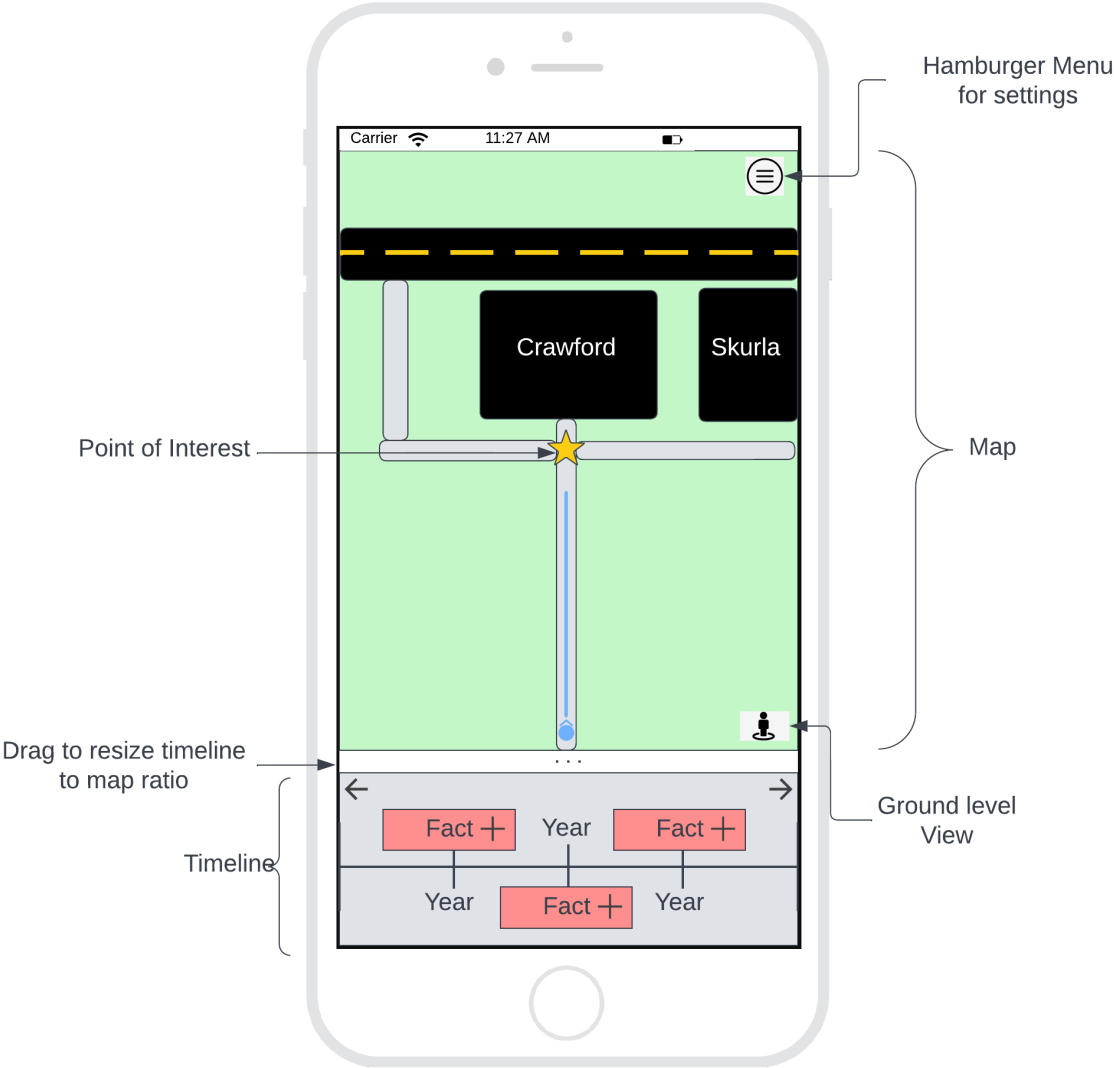
6.2.1 Free Roam



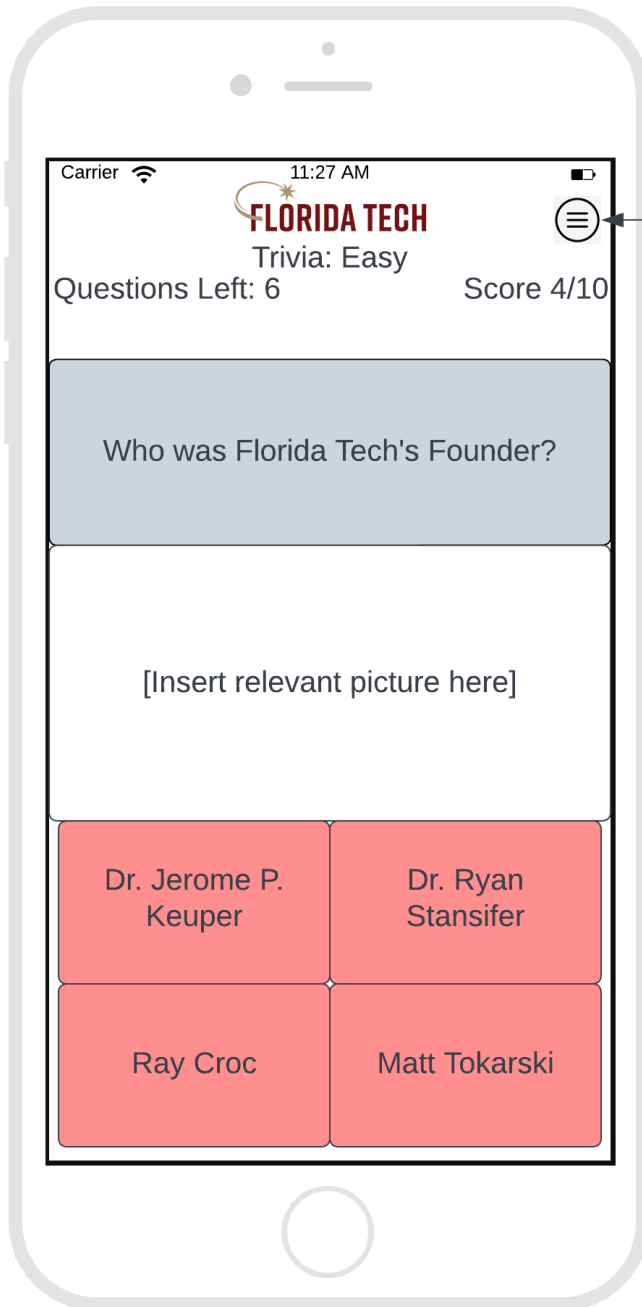
6.2.2 Virtual Tour



6.2.3 Guided Tour

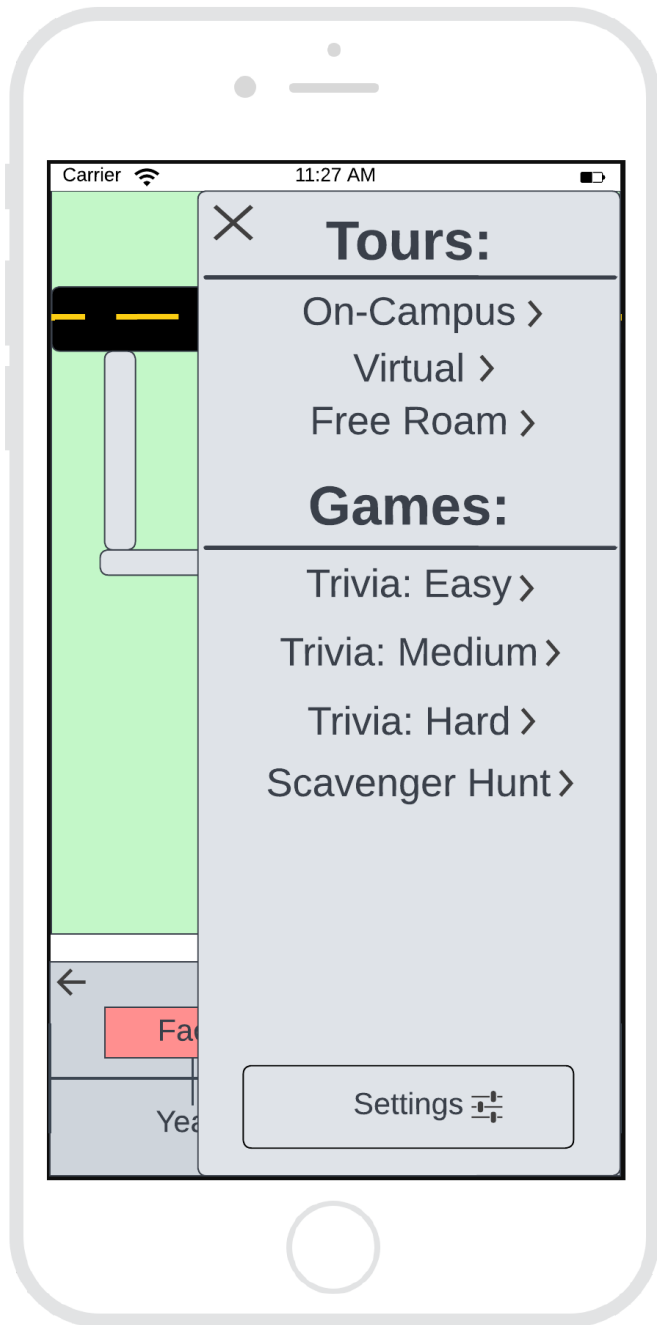


6.2.4 Trivia

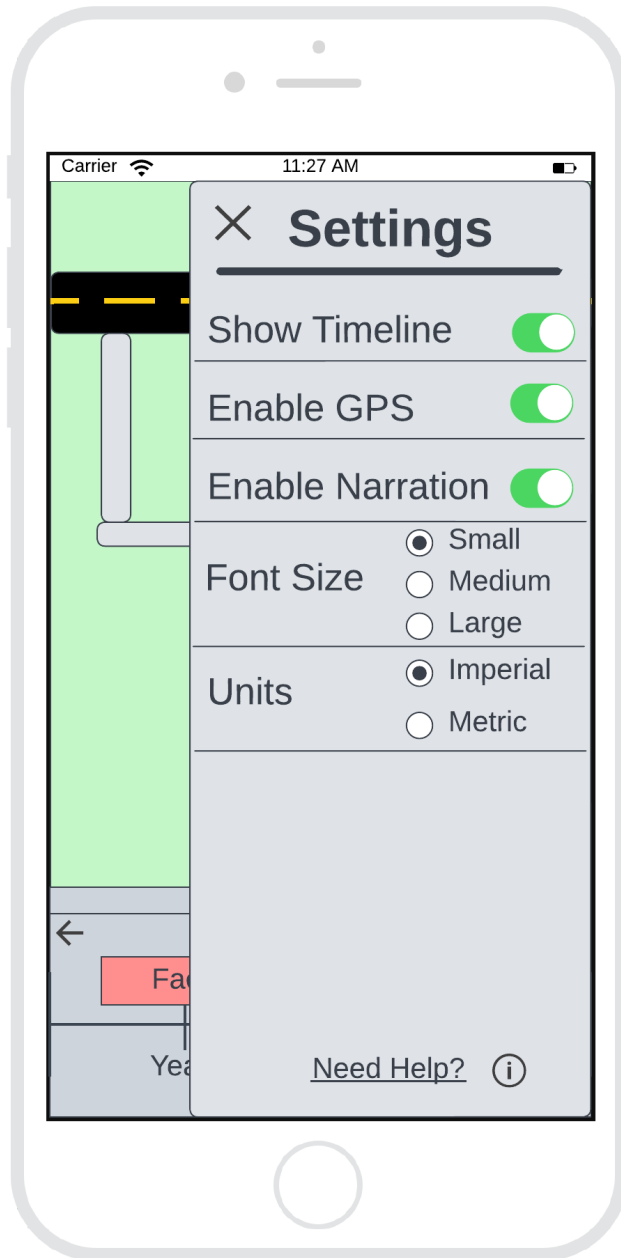


Hamburger Menu
for settings

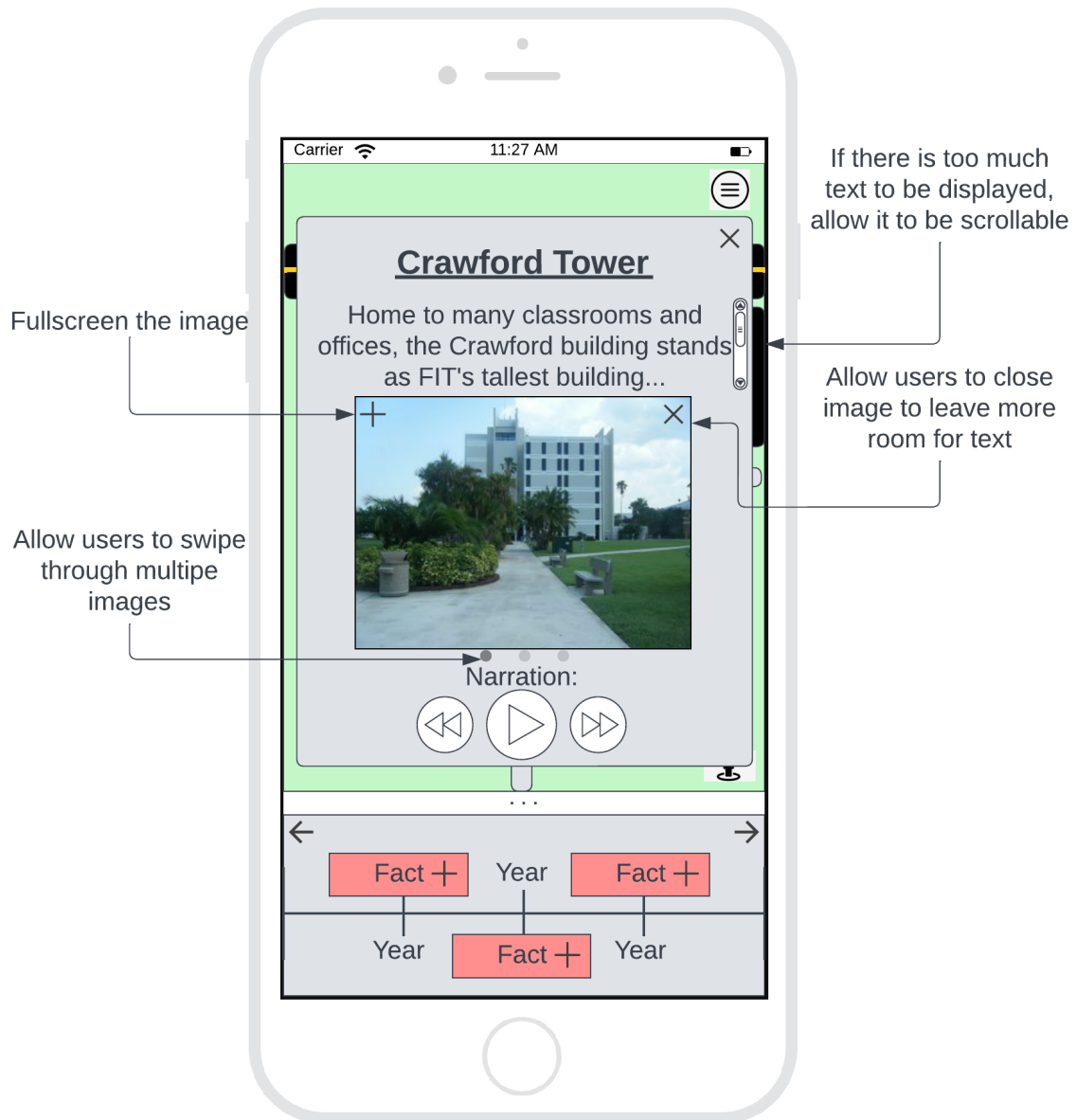
6.2.5 Navigation Menu



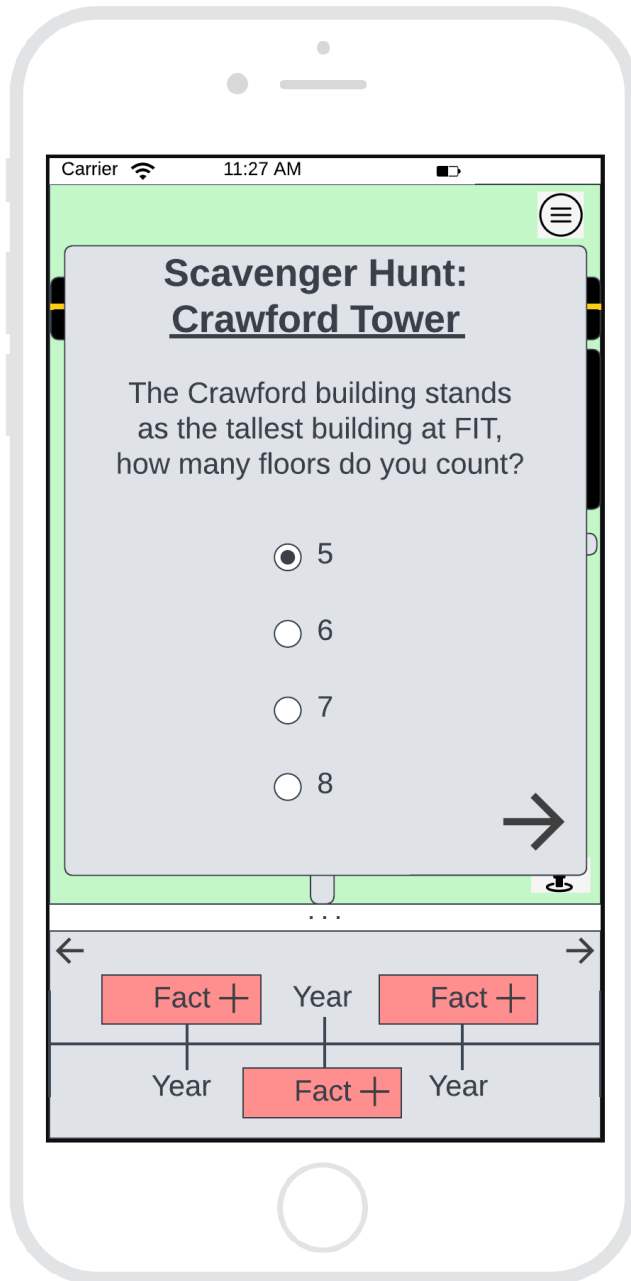
6.2.6 Settings Menu



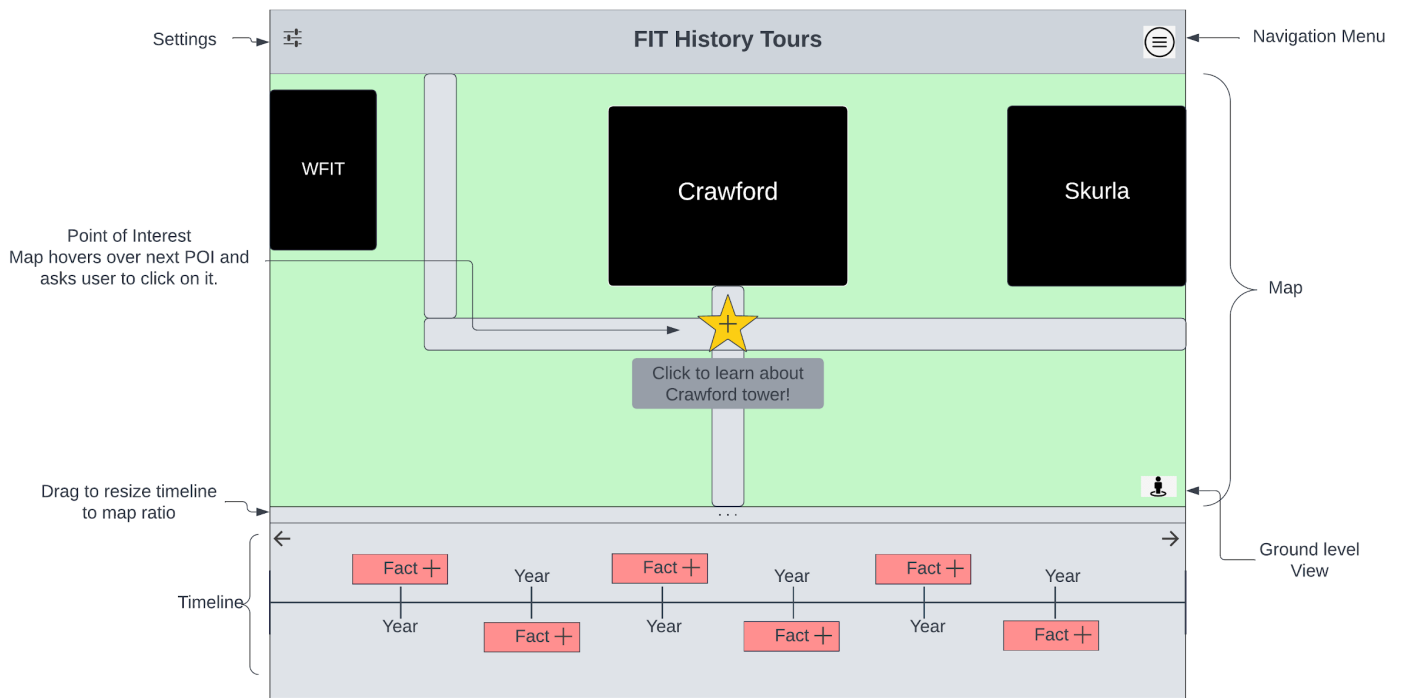
6.2.7 Point of Interest Example



6.2.8 Scavenger Hunt Example



6.2.8 PC Virtual Tour



6.3 Screen Objects and Actions

6.3.1 Map

Prominently the first thing that the user sees, the map shows the user their location in relation to the campus, displaying points of interest, a view of the surrounding area as a map, and different labeled landmarks around them.

6.3.1.1 Points of Interest

Shown as markers on the map, different points of interest are shown to the user. When pressed on, the points show what landmark they belong to, along with information about it. The timeline will respond by showing information solely about that particular point, allowing the user to explore facts about it. During a Guided Tour, these points are used as waypoints that the application brings the user towards.

6.3.2 Timeline

The timeline is at the bottom of the screen, and in its base state it shows different information about the campus and the year associated with the facts being presented. When a point of interest is clicked on, the timeline will automatically sort the entries to only include any related to that point. A larger version of the timeline is available to be explored through the side menu.

6.3.3 Side Menu

A side menu of a selection of sub menus for settings, mode selection and timeline will be displayed, allowing the user to change options about how the application functions.

6.3.3.1 Hamburger Button

A button in a shape with the likeness of a hamburger (≡). When pressed, the side menu will slide onto the screen.