User Stories 4-6 Requirements

# User Story 4

As a user, I want to be able to easily delete saved schedules, in case my schedule changes or I have progressed to a new term with different classes, so that I am not encumbered by schedules that are no longer relevant.

## Functional User Requirements:

1. The system shall allow the user to recall a list of stored schedules by pressing an appropriate menu button.
2. The system shall allow the user to edit or delete any recalled scheduled.

## Functional System Requirements:

1. The system shall be designed such that the option to display stored schedules is easily accessible to the user from either a drop-down menu or side panel.
2. The system shall provide the user with “View”, “Edit”, and “Delete” options once a schedule has been selected.

## Non-Functional Requirements:

1. The system shall store the user’s list of saved scheduled in a single file not exceeding 25MB.
2. The system shall respond to a request to display or edit the saved schedules file within 2 seconds 99% of the time.

# User Story 5

As a user, I want the app to automatically determine the geographical location of buildings that my classes are in based only on the name of the building as provided by the schedule in the TTU registration system, so that I can easily have my schedule generated without research or prior knowledge of where the buildings are.

## Functional User Requirements:

1. The system shall intelligently determine the on-campus location of each of the user’s classes based only on the building prefix and room number entered by the user during schedule input.

## Functional System Requirements:

1. The system shall utilize a data structure/database that maps building prefixes and room numbers with physical addresses on campus.
2. The system shall provide mapped physical addresses to the mapping engine for use during route generation.
3. The system shall generate and display an error message for any invalid schedule input that does not map to a physical address.
4. The system shall generate and display an error message in the event the mapping engine service cannot be reached.

## Non-Functional Requirements:

1. The system shall correctly process all building number to physical address mappings for a given schedule within 3 seconds 99% of the time.

# User Story 6

As a user, I want the app to intelligently determine when classes for the day are already over and provide the reduced set of directions only to classes which are not yet over, so that I am not encumbered by excess, unnecessary information.

## Functional User Requirements:

1. The system shall dynamically generate the user’s route based on the selected schedule and time of day.

## Functional System Requirements:

1. The system shall be able to retrieve the current time of day from the operating device’s onboard system time.
2. The system shall generate a local copy of the selected schedule when the user chooses to generate their route.
3. The system shall cross-reference the retrieved time of day against the schedule’s class times.
4. The system shall remove from the local schedule copy any classes which have already ended.
5. The system shall pass the validated schedule copy to the mapping engine for route generation.

## Non-Functional Requirements:

1. The system shall process a time validated schedule to the mapping within 3 seconds of the user requesting a route, 95% of the time.
2. The system shall not require any further input from the user from the time the user requests a route until the mapping engine returns its results.