

## Sprint #3 Plan

**Product Name:** E-tendance, **Team Name:** Midnight Cobra

**Sprint Completion Date:** 3/13/18 , **Revision Number:** 1.0.0, **Revision Date:** 2/23/18

- **Goal:** Set up the app and database so that a professor/student is able to create an account and their account information is saved in the database. User should be able to sign in to their account once created, and app functionality should be different for professors and students. The geolocation should be implemented in this story so that professors are able to ensure that the students are physically present in class when “checking-in”. Another goal for the sprint includes making the UI more visually appealing in order to attract more users to the app. We want to implement helpful features on the Professor side such viewing who is present/absent per class as well as sending announcements to their students. The students will have these announcements stored on their main class page, ranked by priority and have a viewable record of their attendance .

- **Task listing, organized by user story:**

**User story 1:** As a developer, I want to understand the Google Location Services API in order to implement the Geotcheck functionality into taking attendance. **Story point: 5**

Task 1.1: **Research Location API** (~5 hours): Conduct research on location API. Goal should be learning how to add a map page with respect to a professor’s current location.

Task 1.2: **Research Setting Location Boundaries** (~5 hours): Conduct research on how to set parameter boundaries originating from a point set on the map.

Task 1.3 **Research pulling user’s current location** (~2 hours): Conduct research on pulling a user’s phone’s current location to verify if the user is in the set parameters

Task 1.4: **Set pop-up to enable location services if not already set** (~3 hours): We will need to enable location services for the app, so we will need to learn how to implement this and request access if the app does not have location data permissions.

Task 1.5: **Fix first crash on map access**(~2 hours): Make sure that when checking in with the student location, the app does not crash.

**Total for user story 1:** 17 hours

**User story 2:** As a professor, I want the students’ location data to be verified upon check-in so that students cannot check-in remotely without attending lecture. **Story point: 8**

Task 2.1: **Implement Map into professor side** (3 hours): Be able to view that the location of student is within the preset boundaries when they are checking-in.

Task 2.2: **Set point on map for origin of attendance boundary** (~8 hours): Allow professor to set the point of the lecture and radius in which students are allowed to check in.

Task 2.3: **Student Check within Boundaries** (~4 hours): Conversely, make sure the student can check in validly inside the preset boundaries

**Total for user story 2:** 15 hours

**User story 3:** As a UI designer, I want the app to have visual appeal and intuitive usability so that professors are more likely to use it. **Story point: 3**

Task 3.1: **Implement Color Scheme** (1 hour): Use an appealing color scheme throughout the app in order attract more users.

Task 3.2: **Implement Backgrounds** (~1 hour): Add light background gradients throughout the app.

Task 3.3: **Add more info to main page of each class** (~2 hours): Add more relevant information to display to the user regarding the course. This may include but is not limited to a course description set by the professor.

Task 3.4: **Give Professor ability set course description** (3 hours): Give professors a field in their class pages to set course descriptions. This will require back-end support as well.

Task 3.5: **Add Progress Bar to each page accessing Database** (~1.5 hours): show a loading icon for each call to the database to give more visual feedback to the users. If possible, set delay to show it if it takes

Task 3.6: **Implement swipe delete on student side for Classes** (~2 hours): have functionality so that when user swipes across the classes it will be deleted from the listview and database

Task 3.7: **Implement email verification**(~2 hours):Make sure that the user that signed up has a valid email.

Task 3.8: **Fix bug with professor logout and student login** (~2 hours): Make sure that the functionality of the professor logout and login on the student side of the app is working

**Total for user story 3:** 14.5 hours

**User story 4:** As a professor, I want to be able to view what students are present, absent, and send my students announcements. **Story point: 3**

**Task 4.1: Ability for professors to view present and absent Students** (5 hours): Be able to see in a listview both the present and absent students during that attendance day.

**Task 4.2: Ability to add announcements** (5 hours): add a tab for each selected day in the calendar view that allows professors to create notifications for students such as due dates, test dates, etc.

**Task 4.3: ListView of announcements for professor**(2 hours): The professors are able to view the announcements that were created in a listview.

**Task 4.4: ListView of announcements for students**(5 hours):The students are able to view the announcements that were created in a listview.

**Total for user story 4:** 17 hours

**User Story 5:** As a student, I want to see important announcements from my professor and view my attendance record. **Story point: 3**

**Task 5.1: List view of announcements** (5 hours): Create a view on the main class page that holds all of the professors announcements. We will have the announcements displayed based on priority on what is urgent, important, not important, etc. This will task will require pulling the professor's announcements from the Database and displaying it in a custom list view.

**Task 5.2: Calculate a student's attendance record** (5 hours): Display students attendance record on the main class page, showing how many days they've been present in the quarter and absent.

**Total for user story 5:** 10 hours

- **Team roles:**

**Matthew Dunn:** Product Owner, Q/A, Developer

**Katelyn Suhr:** Developer

**Joseph Aronson:** UX Designer, Developer

**Anisha Tavva:** Developer

**Darren He:** Developer, Scrum Master

- **Initial task assignment:** A listing of each team member, with their first user story and task assignment. This should look like:

**Darren He:** 1.1, 1.2, 2.1, 2.2

**Matthew Dunn:** 1.1, 1.2, 1.3, 4.2

**Katelyn Suhr:** 1.3, 3.2, 3.3, 5.1, 5.2

**Joseph Aronson:** 3.1, 3.2, 3.3, 3.4, 3.5, 4.2

**Anisha Tavva:** 1.1, 1.2, 1.3, 1.4

- **Initial burnup chart:** A graph giving the initial burnup chart for this sprint and is labeled as such with sprint number and project name and is located in the lab.
- **Initial scrum board:** Also known as a task board, the scrum board is a physical board and labeled as such with sprint number and project name and located in the lab. This board has four columns, user stories, tasks not started, tasks in progress, and tasks completed. Index cards or post-it notes representing the user stories and the tasks for this sprint should be placed in the user stories, tasks not started, and tasks in progress columns. Tasks associated with a user story should be placed in the same row as the user story.
- **Scrum times:** List at least the three days and times during the week when your team will meet and conduct Scrum meetings. Also, indicate which of these meetings will have the TA visit as arranged with the TA. It is expected the TA will visit during the Scrum meeting during your lab time.
  1. Tuesday @ 3:30 PM
  2. **TA Meeting: Thursday @ 11:00 AM**
  3. Sunday @ 3:00 PM