# Events Module Development Plan

**Aaron Thompson, Hamilton Turner**

This is an example module development plan. Aaron and I will help to make a similar plan for each module group. The key features are the user stories, and the timeline. Plans can also include long term goals for the project (things to possibly do in the future), scheduled peer code reviews, refactorings(code and/or user interface), demonstrations, and meetings (besides the standard mentor meeting after each build cycle).

## User Stories for Events module

This should serve as an example of how user stories are laid out. They all relate back to the end user, keeping the focus of the project not on the technical details of the programming, but on the end results of that programming and the main goal. Notice that they all start with "user can ..." and we have tried to keep them as action-oriented(verbs!) as possible. Also notice that there is nothing in here about programming, and these are just as valid for iPhone as for Android. These users stories will change a bit, but not much. For whatever module you decide to work on, Aaron and I will meet with you to come up with your own user stories.

Users can:

- Submit events through web interface with information such as location (building name, possibly room, translated to lat/lon upon reaching the server), date, time, target audience and other meta data.
- Submit events using the Events module -d

- Access information pulled from university sites like Common Place and the Calendar.
- Access events pulled from Social Media Services such as Facebook and Twitter

- Filter events based on time Ex: right now, tonight, tomorrow, this weekend -d
- Filter events based on location Ex: My location, Featheringill, Commons -d
- Filter events based on meta data provided (tags). Ex: free food, live music -d
- View / Modify their current filters

- View a map overlay of events matching their current filters -d
- View a list of events matching their current filters -d
- Elect to be notified when passing in close proximity to an event venue -d

- View aggregate scores of events currently in progress
- Users can submit simple ratings (up/down) of events while attending them
- Submit small messages regarding an event (they ran out of food, band is done playing, etc)

- Associate an event with a Facebook event
- Invite other students to the event through the app, displaying a push-type notification on the receiving students device
- Add the event to the device calendar

- Edit events which they created
- Edit events using a web interface
- Invite others to be able to edit events

## Timeline for Events module

This should serve as an example of how a module timeline is laid out. The development model we are working with uses 'Build Cycles,' during which you choose the user stories you plan to accomplish. Note that the time you give yourself includes everything needed to complete those user stories – deciding how to implement, implementing, testing, etc. At the end of the build cycle, except for rare cases, the user stories should be entirely complete. Aaron and I will help you plan your timeline, and help balance the builds, so that each cycle has a similar amount of work. We recommend a 2 week build cycle. See http://code.google.com/p/javassonne/wiki/User_Stories for some more examples. Also, many of your module groups will have more than one person. In that case, assign individuals to specific user stories (see the example link). For inexperienced developers, Aaron and I will be sure to ensure that your first few user stories are realistic. For example, many of you may have the 'user can see application icon on home screen', or 'user can launch application' stories. These are a good way to get started and involved in the programming.

1st Build (Sept. 9 – Sept. 19)

- Submit events using the Events module
- View a map overlay of events matching their current filters
- Filter events based on location Ex: My location, Featheringill, Commons
- View a list of events matching their current filters
- View / Modify their current filters

Sept 21 – group check in with mentor
2nd Build (Sept. 20 – Oct. 3)

- Filter events based on meta data provided (tags). Ex: free food, live music
- Filter events based on time Ex: right now, tonight, tomorrow, this weekend
-

Oct. 4 – group check in with mentor
3rd Build (Oct. 4 – Oct. 17)

- Submit events through web interface with information such as location (building name, possibly room, translated to lat/lon upon reaching the server), date, time, target audience and other meta data.
- Access information pulled from university sites like Common Place and the Calendar.
- Access events pulled from Social Media Services such as Facebook and Twitter

Oct. 18 – group check in with mentor
4th Build (Oct. 18 – Oct. 31)

- View aggregate scores of events currently in progress
- Users can submit simple ratings (up/down) of events while attending them
- Submit small messages regarding an event (they ran out of food, band is done playing, etc)

Nov. 1 – group check in with mentor
5th Build (Nov. 1 – Nov. 14)

- Associate an event with a Facebook event
- Invite other students to the event through the app, displaying a push-type notification on the receiving students device
- Add the event to the device calendar

Nov. 15 – group check in with mentor
6th Build (Nov. 15 – Nov. 28)

- Edit events which they created
- Edit events using a web interface
- Invite others to be able to edit events

Remember that encapsulated within a single user story are multiple issues. For example, take the user story “Associate an event with a Facebook event.” To finish this user story, you need to decide on an interface on the device. Where will the 'associate with a facebook event' link go, and what will it look like? Where will the association be stored?  Is is part of the event meta information? Can this be done only when the user creates an event, or can it be done after the fact?

Be sure to remember that smaller devices means screen space is more valuable. Good, logical organization of menus and options can be the key to a good application. Also, before spending time on development of a feature, make sure that you consider whether or not that feature is actually important to the end user. Even with user stories, which try to alleviate this problem, 'feature creep' is still an issue.