Partí Application Documentation

Justin Fern, Kyle Huntsman, Brandon Olson
Advancing Computer Science Department, University of Advancing Technology
2625 W Baseline Rd, Tempe, AZ 85283, US
jusfern@uat.edu
kylhunts@uat.edu
braolson@uat.edu

I. Feature Driven Development

Feature Driven Development (FDD) is an agile process that focuses on the features of the end product. The features are small and focus on the features the users will find useful.

A. *A Developed Model*

We would need to develop a model of the application. This model shows what the application is, what it'll do, and how it will achieve what it needs to do for the user. We have achieved this by creating use case diagrams, and an overall description of what our application accomplishes. However, we still need to develop a class model.

B. *Feature List*

Next, we would have to create a feature list. These will be user-features, and must be broken down into the smallest features possible that the user will still find useful. For example, instead of a feature being a search bar, the feature might be clicking the search button. The next feature would be starting search, etc. Each of these small features can now be used as individual sprints.

C. *Using Feature Driven Development*

Using FDD allows progress to be shown in a user orientated way. Shareholders can easily view progress with the application when they can physically interact with the application and visually see the difference.

II. Documentation Of Time

Each week will be defined to include what word was done that week and how much time was included to complete each task.

A. *Week 7*

This week the team worked creating a forum to make Parti invitations and an expandable card view to represent those invitations. In a previous work session the team had already created a basic tabbed layout which included three different tabs of public, private, and rsvp. Each tab displayed a list of card with simple placeholder information within them. To start we created cards that were able to expand and contract when they were clicked and a forum to create events. Most of the time this week has been focused around tweaking these two features. Towards the end of the week the team found that there was an error in card view and we spent time trying to resolve the issue. Lastly, we also started to convert the project over to fragments. For the most part the time spent on tasks looked like following:

Task Name	Date	Duration (hr)
Updated expanding card view	10/23	3
Parti creation form	10/23	3
Converting the project to fragments	10/23	1

B. *Week 8*

This week was not very productive content wise for the project. The team mostly assembled to discuss the design and direction of the application. The only thing that the team was able to complete this week was that we converted the Event Creation Fragment to an activity because it was a part of a different workflow.

1) *Goals:*

- Describe the current direction of the project
- Decided how we want the UI to be interacted with
- Change the Event Creation fragment to an Activity

2) *Challenges: This week the main challenge was that we were not able to meet as a team as often as we wanted. Because of the lack of planning and time this week were not able to get as much done as we would have liked.*

3) *Accomplishments: The only main achievements this week is that we learned more about Android development and we changed the Event creation fragment into an activity.*

4) *Next Time: Next week we are going to get allot more done. At the beginning of the week we will make a formal features list and attempt to get as many features as we can completed by the end of the week.*

C. Week 9

This week we made progress in both design and application. We decided to focus on how each tab with the main activity of the application was going to function. We planned how to fill each fragment's contents and how we wanted the cards to look.

Additionally, we created the flow for the user to access the Event Creation Activity. This flow went through several revisions. Firstly, we added a new tab to the swipe tab menu. After giving it more thought, we decided to access the tab using a Floating Action Button (FAB) located on the 'HOSTING' section of the swipe tab.

1) *Goals:*
   - Access the Event Creation Tab through the current UI
   - Revise card's look and feel
   - Find a way to fill each card within the fragment's dynamically
   - Fix card Action button padding changing when card is collapsed/reopened

2) *Challenges: This week's list of challenges is quite lengthy. Firstly, we never got the card action button error fixed. Secondly, creating the FAB took more work than we originally expected. Thirdly, when setting the on click listener for the FAB, the app was crashing. Why? We still don't know. When starting the Event Creation activity, the app also crashes. This needs to be fixed.*

3) *Accomplishments: We gave serious thought into how fragments should be laid out. We created a FAB successfully. The Event Creation fragment was also created.*

4) *Next Time: Finish the Event Creation activity. The team will begin development on sending invitations to other app owners.*

D. Week 10

The week the team has finally become very organized. We decided to divide tasks between each of the team members instead of peer programming of each of the tasks. To make the distribution of tasks equal we gave each team member a feature level tab. The distribution went as follows, Brandon worked on the Hosting tab, Kyle worked on the Invites tab, and Justin worked on the RSVP tab. This was also the most productive week as allot of features were completed. The application now has the special cards for each of the tabs, ground work for creating events, ground work for inviting people to events, and lastly the ground work for accepting and declining events.

1) *Goals:*
   - Separate the current work to everyone on the team
   - Get work done before Sunday
   - Get allot of work done

2) *Challenges: The week was challenge free. There was not really anything that got in the way of being able to get work done. We could have used more time in the week but that is always true.*

3) *Accomplishments: We made a significant amount of progress and now have a workflow that the team can use to get allow of work done on the project.*

4) *Next Time: We will continue our leftover work then assign new tasks to each of the team members.*

E. Week 11

This week, the team probably worked for 15+ hours on the project. This week we focused on the backend database logic for app, as well as researched common programming patterns for managing database operations.

1) *Cupboard: To help manage the local database on the phone, we're using an API called Cupboard, which makes the process of storing and retrieving POJOs to and from a database super easy. Here's a link to the Cupboard API site.*

2) *CRUD: To make things easier on ourselves, we implemented CRUD operations for our database objects. CRUD stands for create, retrieve, update, and delete. These four operations outline the major functionality when needing to store and retrieve information to and from a database. Our CRUD API allows us to easily retrieve and store our objects to and from our local database using Cupboard, and to and from our remote database using GET and POST requests to the PHP scripts on the server.*

3) *PHP Scripts: The PHP scripts on the remote server were finished. We can now all CRUD operations to and from the remote server. The PHP scripts both take and return JSON formatted strings. This allows us to use Android's JSON parsing library to pass information to and retrieve information from the remote server. GET and POST requests*

to these PHP scripts are written in Android's AsyncTask class, as internet operations should not take place on the UI thread.

F. Week 12

This week was all about restructuring and merging all the branches of the project. At about week 10 the team decided to divide all of the tasks of the project into their own branches. This dividing of features worked well at first but later into development the team found a couple of issues with having long running branches that have never merged together. We were able to merge all the branches into the development branch and fix all of the merging errors that had occurred. Additionally, the team worked on getting the event creation feature fully implemented in to the application. The team worked on hooking up the connection to the database when an event is created. The team also setup date and time dialog pickers for the user to set when an event starts and ends.

1) Goals:

- Merge all Branches
- Build Create Event Feature

2) Challenges: One challenge that we overcame was merging the branches. The project's feature branches had been apart for a long time. Because the branches had been apart for so long we as the developers had to heavily edit the same file to get the results we needed. Additionally, we would sometimes need features to be created that we all needed to continue. So we decided to reset all of the branches to development and build off from there again.

3) Accomplishments: The team learned allot more about version control branch management after merging several feature branches into the development branch. The team also learned how to use time and date dialogs.

4) Next Time: Next week is all about synchronizing all of the applications features. Currently, all of the features keep to themselves and don't talk to each other or the database. We hope to resolve this issue in the feature

G. Week 13

The project is entering its final stages this week and the project's features are starting to come together. The group mainly worked on getting the current features to be able to interface with the database. Additionally some new features made it into the project this week. When an event is created by the user it now is saved to both the external and local databases. Additionally, users are now able to edit events that they are hosting. Any changes that are made to the hosted events will be reflected in the external and local database. In the future the team will implementing the ability for the application to notify all the attendees of any changes made to a hosted event. For example, if the time or date of the event changes then all of the attendees will get a notice.

1) Goals:

- Have user interaction between both external and local databases
- Improve the graphical user interface for event creation and hosting cards

2) Challenges: The most challenging part of this week was updating the events that were in the external database. The team spent a long time try to determine why the new event was not updating on the external database. In the end we found out that the code was send the unedited version back to the database for updates and all we had to do was send the edited version.

3) Accomplishments: The team was able to finish viable user to database interaction.

4) Next Time: Next time the team will add more features such as inviting users to events via texts or email. Also the team wants to change the user interface to both increases productivity and graphical validity.