Deliverable 1

# Project Title:

DiscoEvents

# Group Name:

Novus

# Group Members:

Brandi Werner, Tsung-Han Hsieh, Daniel Jimenez, and Kaytlin Lafleur

# Project Description:

The system that we are developing will allow Engineering Organizations and UNT Departments to submit their events through a web page. In order to do this the Organizations/Departments will need to create an account on the webpage. By creating an account the organizations/departments will be creating an information page about their group. This information page will include their name, a description of what they do, their contact information, links to their different social medias, and a picture of their choice. After they create an account they will be able to submit an event. The event application asks for the date, details, description, major/s the event covers, and a picture for the event. Once the event is submitted or edited, it will need to get approval by an administrator of the website before it will be posted to the app.

The next part of our project is an app that displays the events that the organizations/departments submitted. This app is specifically made for the purpose of tablets being set around Discovery Park that students will be allowed to access and see what events are happening. Students will also be able to download the app onto their personal devices. This app will have all of the events displayed on a calendar view. It will also have a menu where the user can look at the different events by categories. These categories will be the major or department the event is associated with. For example, if the event is a cybersecurity event, it will be under the category of Computer Science. We plan on implementing a search bar where the user can search the for events through key terms. If the app has not been in use for 3 minutes we plan to have it run a slideshow of all of the events.

The software we will be using to create our website is Django. Django is a python based web framework that allows developers to develop rapid and clean designed websites. This means that we will be using the languages Python, HTML, and CSS when creating our webpage on this framework. A benefit to Django is that is take many steps to secure your websites.  As inexperienced web designers, this is a bonus because we do not know all the tricks to securing a website. Django also has a database that it’s websites use. This will be useful to us because we need a database to store the events that will be displayed on our app. With this database we will be able to use the Django Rest framework to send data to our android app.

To build our android app we will be using android studio which uses the languages Java, and XML. We selected android studio because it has many plug-ins that we can use and has many online references and tutorial if we need help. One of those plug-ins is a calendar plugin. Android studio also has the capability to send request to our Django website and pull those request back. Once our app has the data it needs, we can easily use android studios to format the information the way we want.

Our application will have many benefits for both the students and the organizers of events at Discovery Park. The students will be benefitted by having a centralized location of all the events that are happening at Discovery Park. They will also be able to discover different event that may pertain to their interest or major. By having an easy and accessible way for the students to find out what different events are happening at Discovery Park many more students will be able to participate in events which will help the organizations/departments. Organizations/Departments will also benefit from this app because they will have one more way to promote their events to students.

# Meeting Minutes:

9/5/17 Meeting

Location: class

Present:   
Brandi Werner, Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Deliverable 1  
Github group structure  
Presentation preparation

Deliverable 1 - discussion was mainly to split the content of deliverable 1. We have each chosen to do either one or two parts from the deliverable, and the rest will be shared. Kaytlin has taken charge of the powerpoint information and creation. We will each brainstorm 3 aspects of risk management to share at our next meeting. App working title is DiscoEvents.

Github group structure - we decided to keep our github on the public version of the application. We added the professor and the TA to our github. We discussed the github structure as well as the accessing policies. We decided to pull the full android development folder and upload it when done. This will ONLY be done if notified prior on Groupme

Presentation Preparation: Group member Kaytlin Lafleur has chosen to handle powerpoint presentation. She will do this based on the finalized deliverable 1.

9/7/17 Meeting

Location: class

Present:   
Brandi Werner, Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Timeline of events   
Discuss risk management  
Discuss project scope  
App format  
UI plan  
Data handling  
Deliverable 1 progress

Timeline of events - we discussed our deadlines, and milestones. We built a schedule to go off of, and Daniel will create a gantt chart based off it it.

Discuss risk management - at our last meeting we each decided to brainstorm 3 risk management scenarios for this meeting, and we would keep the best 5. We compared all 5 of our risks and decided to stick with: non-compatibility, scope ill defined, bad communication, losing a member, lack of experience with android/django.

Discuss project scope - we decided a brief project scope and have adopted a two part application, with a web app and an android app. The format will be decided at a later time, when we have more of a handle of our programming abilities. For data handling, our web app will organize a database for us, that the android app will pull.

Deliverable 1 - we touched bases to ensure that the deliverable and its aspects are on track. At this point the deliverable is nearly finished as we have all split the parts of the deliverable. What is left is compiling the information and fine tuning.

9/12/17 Meeting

Location: class

Present:   
Brandi Werner, Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Deliverable 1 progress  
Group member roles  
Milestones

Deliverable 1 progress - we analyzed our deliverables strength and what was left to be done. We decided to add more to the schedule section. Also we must change the group roles section as we misunderstood the outcomes.

Group member roles - seeing as we misunderstood the roles section of the deliverable, we each discussed our roles. Daniel J is working on android and android UI. Brandi is working on android and is the project lead. Tsung H is working on Django web programming, and Kaytlin is working on Django web programming and overall UI design.

Milestones - we looked at our schedule to re-evaluate our tasks with help from the professor. We wrote out milestones that each had a corresponding deliverable to expand on in the final deliverable 1.

9/12/17 Meetings

Location: Library

Present:   
Brandi Werner, Kaytlin Lafleur, Daniel Jimenez, Tsung-Han Hsieh

Agenda:   
Presentation parts  
Group member roles

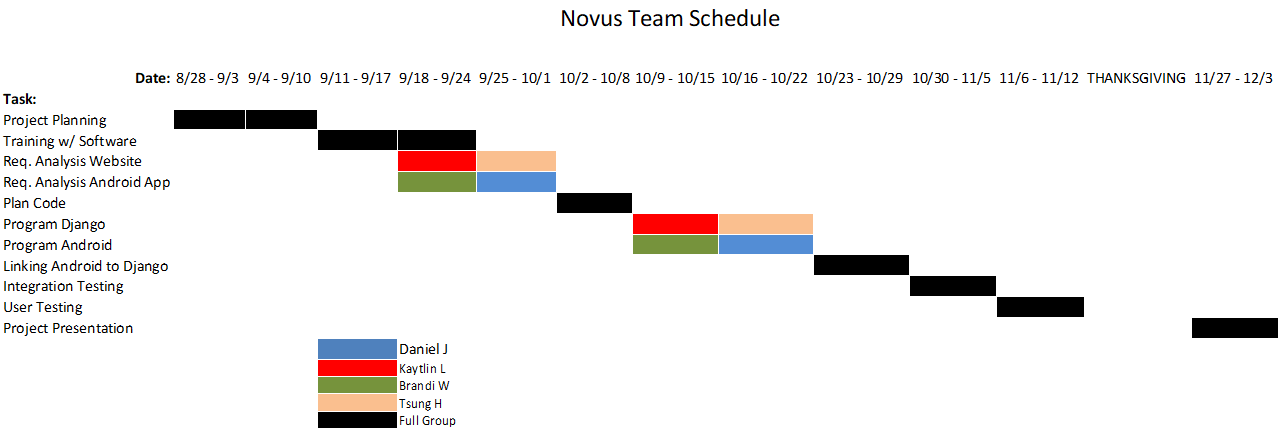
Presentation parts - this meeting was primarily focused on the presentation of deliverable 1. We reviewed the slides and split up each section according to what we worked on in the deliverable. We also did a practice presentation to ensure we meet the allotted time for the meeting, and also met group member with constructive criticism for presentation improvements.

Group member roles - one last look was given to create more detailed member roles, which will be included in deliverable 1.

# Initial Planning:

Milestones - our milestones listed below are each critical tasks to complete for our project. They are each paired with their respective deliverable.

* Project planning: the deliverable will be this project report, Deliverable 1.
* Requirement analysis for web page: We will analyze our requirements and scope for the web page, and our deliverable will be a plan for how we will create the webpage, as well as its functionality.
* Requirement analysis for android app: We will analyze our requirements and scope for the android app, and our deliverable will be a plan for how we will create the android app, as well as its functionality.
* Programming webpage: By this time we will deliver a web page created using Django framework as a deliverable.
* Programming android app: By this time we will deliver an android app as a deliverable .
* Testing and Debugging: By this time both the Django and Android apps will be developed and we will be testing their compatibility and for any glitches.



# Risk Management:

Below are our top five risks managements. We selected these five because we felt that they are the risk that have the most probability of affecting our group due to both our experience level and none of our team members working together previously on a project.

1. Lack of experience with android/Django
   1. The way we will monitor our lack of experience is through setting goals on the different sections of applications we are working on. If we reach these goals in a timely manner then we will know that our lack of experience is not affecting our ability to work on our project. If we miss our deadline for one of our goals, then we know we will need to work harder to overcome our lack of experience.
   2. As the semester progresses, we will evaluate our progress on the project to see if our lack of experience is affecting us in a negative way. Once we have done this we will look at what is left to accomplish with the project and determine if we are unexperienced in the parts of the project we still need to do.
   3. We will allocate slack days and/or utilize group meetings as extreme catch up days. We will also each individually research the item we do not know how to do before we need to program those parts of the code.
2. Losing a member
   1. We will be checking on who have attended to classes. If someone has been consistently being absent then we will know to split the project into bigger chunks.
   2. As the semester progresses, we will check who has or hasn’t been attending. Then we can see how much work in the semester we still need to do and adjust accordingly.
   3. other members will have to work more to catch up with the schedule.
3. Bad Communication
   1. We will find out if a bad communication has occurred after the task is finished or when we meet, if there’s any misunderstanding then we will know to explain things better.
   2. If the bad communication had greatly impacted the progress, then we’ll speed up and get it right the next time to make up for the lost time.
   3. specify and explain the task more clearly in a more understandable way.
4. Scope is ill defined
   1. We will stay in constant communication with each other about what we are working on so all of us are on the same track about what we are doing.
   2. If we still are having problem with our task and understanding what our project is and our different contributions, then we will devote a whole meeting to this task.
   3. We will work on the project all at one time instead of individually that way each of us is available to answer questions another might have.
5. Non-compatibility
   1. Firstly, we are going to research how and whether we can connect Django to android studio. Secondly, for users who do not have an adroid app we’ll check on how many times that someone tries to use the app but couldn’t because of the non-compatibility.
   2. We will research and ask our friends who have experience with Django and Android Studios to see if they can help us figure out how we can solve our problem. If the non-compatibility has become a big issue to the users who do not have an android, then we might raise our priority on making it accessible to more platforms
   3. For the first compatibility problem, we will create our webpage by hand and connect it to a database system such as MySQL since we know android studio is compatible with that. Secondly, we will make a port that is accessible with more platforms.

# Progress on Deliverable:

The way we created this deliverable was by breaking up the requirements and giving each person several requirements to accomplish. By doing this we were able to distribute the work evenly and give every person in our group a fair amount of tasks. Also, by having only one group member work on specific tasks they were able to research what was required for that task and become an expert on it. Once, becoming an expert on this topic they are able to teach the other team member what they have learned. This lead to every team member becoming an expert on all the topics while not every team member having to research all the topics on their own.

Through creating this deliverable we were faced with a lot of questions such as:

* What are the key features we want our app to do?
* Which software will be best to use for our app?
* Which software will be best to use for our website?
* How will we deal with several people working on the same code at once?

Since many of our team members are new to building apps, we had to do a lot of research to answer these questions. Through our research we learned about the applications we will use, what exactly we need to do, and approximately how long each task will take. Knowing these things we are able to set goals for ourselves.

Our next task after creating this deliverable will be to accomplish our goals. This will not be an easy task since we have to learning in depth about the new languages and systems we will use. If we do not accomplish this in a timely manner then our whole project’s timeline will be pushed back. Thus, form this deliverable we have derived that our current main goal is to learn the new programming languages and systems as fast as possible.  Another one of our main goals is to do enough research and plan out pseudocode well enough that we will run into minimum amount of problems.

This deliverable really made our group think about how we would accomplish our project. Through describing what we wanted our app to do and how we are going to accomplish our task, we had to do a lot of research. This research was very informative about all the different ways we could build our project.

# Roles of Members:

Brandi Werner: Android Developer, Integration Developer, Project Lead

Daniel Jimenez: Android Developer, Integration Tester, Documentation

Tsung-Han Hsieh: Django Developer, Error Tester (Create the testing environment), Documentation

Kaytlin Lafleur: Django Developer, Visual Designer, Presentation Lead

# Repository Policies/Steps

The following project repository checkout and update policies are created so that when multiple people are working on the code there will be a limited amount of conflicts when committing the code back to GitHub. The following steps should always be followed in the order given.

1. Let the group know on GroupMe that you will coding, what code you are pulling, and the changes you will be making to the code.
2. Create a branch from the master code. This essentially make a copy of the master code. A team member should never make direct changes to the master code.
3. Checkout the new branch that you have just created.
4. Make the changes to the code that you need to make. Test the code and make sure it works properly.
5. Check the status of your branch to make sure that your changes were saved. If the status of your branch is “modified”, then move on to the next step.
6. Use the “Add -A” command to add all of your changes.
7. Commit your code using the “commit -m”.  Make to leave a detailed comment when adding your changes. The comment should describe exactly what you did and what line of code you changed or added.
8. Create a pull request for your branch. What this does is send your code up to GitHub so other people can see your changes and peer review them.
9. Checkout the Master Branch to make sure that none of your changes are conflicting with changes another person has made to the code while you were working on it.
10. “Pull” the master branch to see if it was updated.
11. If the master branch was updated during the times you were making your changes, checkout your new branch again. Then merge your branch and the master branch. What this will do is try to merge the master branch to your branch. If you can, try to rework the code so both your new code and the other updated code are both in the program.
12. If you cannot rework the code then let the team know on group chat and further action will be taken to solve the problem.
13. If you can rework the code, rework the code to make it not conflict. Then commit and push your branch.
14. Go to the Project’s repository on GitHub and create a pull request. By doing this you asking another team member to review your changes to the code and merge the code. You must never merge your own code.
15. Let the team know on GroupMe that you are finished with your code and that you would like someone to review your pull request.

# Member Contribution Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution (%) | Note  (if applicable) |
| Brandi Werner | Project Leader, Android Developer, Integration Developer | 25% |  |
| Tsung-Han Hsieh | Django Developer, Error Testing (Create the testing environment), Documentation | 25% |  |
| Daniel Jimenez | Android Developer, Integration Tester, Documentation | 25% |  |
| Kaytlin Lafleur | Django Developer, Visual Designer, Presentation Lead | 25% |  |