Deliverable II

DiscoEvents

Novus

2017

Part 1

Software Requirements Specification

**for**

DiscoEvents

Novus

Table of Contents

[**1 | Introduction** 1](#_Toc495437351)

[1.1 | Purpose 1](#_Toc495437352)

[1.2 | Document Conventions 1](#_Toc495437353)

[1.3 | Intended Audience and Reading Suggestions 1](#_Toc495437354)

[1.4 | Project Scope 1](#_Toc495437355)

[1.5 | References 2](#_Toc495437356)

[**2 | Overall Description** 2](#_Toc495437357)

[2.1 | Product Perspective 2](#_Toc495437358)

[2.2 | Product features 3](#_Toc495437359)

[2.3 | User Classes and Characteristics 3](#_Toc495437360)

[2.4 | Operating System 3](#_Toc495437361)

[2.5 | Design and Implementation Restraints 4](#_Toc495437362)

[2.6 | User Documentation 4](#_Toc495437363)

[**3 | System Features** 4](#_Toc495437364)

[3.1 |Android App Features 4](#_Toc495437365)

[3.1.1 | Home Screen 4](#_Toc495437366)

[3.1.2 | Calendar 5](#_Toc495437367)

[3.1.3 | Events of the Day 5](#_Toc495437368)

[3.1.4 | Organization List 6](#_Toc495437369)

[3.1.5 | Organization Profile 6](#_Toc495437370)

[3.1.6 | Search Page 7](#_Toc495437371)

[3.1.7 | Search Results 7](#_Toc495437372)

[3.1.7 | Event View 8](#_Toc495437373)

[3.2 | Django Web-based Application 8](#_Toc495437374)

[3.2.1 | Login Page 8](#_Toc495437375)

[3.2.2 | New User Sign-up 9](#_Toc495437376)

[3.2.3 | Organization Information Page 9](#_Toc495437377)

[3.2.4 | Create/Edit an Event 9](#_Toc495437378)

[**4 | External Interface Requirements** 9](#_Toc495437379)

[4.1 | User Interfaces 10](#_Toc495437380)

[4.2 | Hardware Interfaces 10](#_Toc495437381)

[4.3 | Software Interfaces 10](#_Toc495437382)

[4.4 | Communications Interfaces 10](#_Toc495437383)

[**5 | Other Nonfunctional Requirements** 10](#_Toc495437384)

[5.1 | Performance Requirements 10](#_Toc495437385)

[5.2 | Safety Requirements 11](#_Toc495437386)

[5.3 | Security Requirements 11](#_Toc495437387)

[Database Security Requirements 11](#_Toc495437388)

[User Identity Authentication 11](#_Toc495437389)

[5.4 | Software Quality Attributes 11](#_Toc495437390)

[**6 | Other Requirements** 12](#_Toc495437391)

[6.1 | Database Interface 12](#_Toc495437392)

[**Part 2 | UML Documents** 14](#_Toc495437393)

[Class Diagram Android 14](#_Toc495437394)

[Class Diagram Django 14](#_Toc495437395)

[Sequence Diagram Android 15](#_Toc495437397)

[Sequence Diagram Django 16](#_Toc495437398)

[Use Case Android 16](#_Toc495437399)

[Use Case Android Failure 17](#_Toc495437400)

[Use Case Django 17](#_Toc495437401)

[Use Case Django Failure 18](#_Toc495437402)

[**Part 3 | Test Plan** 18](#_Toc495437403)

[Test case 1: 18](#_Toc495437404)

[Test case 2: 18](#_Toc495437405)

[Test case 3: 18](#_Toc495437406)

[Test case 4: 18](#_Toc495437407)

[**Part 4 | Updated Risk Management** 18](#_Toc495437408)

[Lack of experience with Android/Django 19](#_Toc495437409)

[Losing a member 19](#_Toc495437410)

[Bad communication 19](#_Toc495437411)

[Conflicting code 19](#_Toc495437412)

[Planning problems 19](#_Toc495437413)

[**Part 5 | Updated Project Plan** 20](#_Toc495437414)

[**Part 6 | Meeting Minutes** 20](#_Toc495437415)

[9/21/17 20](#_Toc495437416)

[9/26/17 21](#_Toc495437417)

[9/28/17 21](#_Toc495437418)

[10/5/17 22](#_Toc495437419)

[**Part 7 | Progress Report** 22](#_Toc495437420)

[**Part 8 | Member Contribution** 23](#_Toc495437421)

# 1 | Introduction

## 1.1 | Purpose

The product being discussed in this SRS is Disco Event App release version 1 as this will be the first release of this software. This software will have two interfaces, a web interface used by organizations and departments and an android interface used by students. The web interface will have the capability of users creating profiles and events for their organization while the android interface will allow students to view event on a calendar view or search for specific events through tags.

## 1.2 | Document Conventions

This document uses MLA Format. Bold face text indicates a Heading,

## 1.3 | Intended Audience and Reading Suggestions

This document is intended for project managers, team developers, testers, and document writers to read because it will help them in understanding the requirements of the software. The rest of this SRS contains an overall description, the systems' features, external interface requirements, and other requirements. It is suggested to read this document beginning with the overview sections and proceeding through the sections that are most pertinent to each reader.

## 1.4 | Project Scope

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 for the 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 goal of our application is to make it easier for user to find events that are related to their interest and help organizations promote their events. The benefits of this application is that student will easily be able to search for events through their phones. This will help Engineering students become more active in their schools and help the turn out rate to organization's events.

## 1.5 | References

Flores, Jose. “Calendar Application (Class Diagram (UML)).” Creately, Cinergix Pty. Ltd, 2013, creately.com/diagram/example/hr0vxdcb/Calendar%20Application.

Used as a reference when creating the Android class diagram.

NikhilRcop. “Sequence Diagram for Android Application That Uses NFC.” UML - Sequence Diagram for Android Application That Uses NFC - Stack Overflow, 26 Apr. 2015, stackoverflow.com/questions/29882661/sequence-diagram-for-android-application-that-uses-nfc.

Used as a reference when creating both the Android and Django sequence diagrams.

To create all of the UML diagrams this SRS used the in class lecture notes as the basis for what each different component of the diagrams are and how they are used. From this we used the above resources help specify our documents to the development tools we are using.

# **2 | Overall Description**

## 2.1 | Product Perspective

Disco events is aimed to be a more accessible and easier to navigate version of OrgSync. DiscoEvents is an event tracking app specifically for UNT engineering students at Discovery Park. We, Team Novus, saw a need that required to be fulfilled for all students. Discovery Park is a large, and diverse campus filled with opportunities for students. There are often multiple events a day at Discovery Park. Some may be run by departments, other by organizations. DiscoEvents is meant to be the mobile solution to being involved at Discovery Park by serving as a hub to display all the events going on any given day at Discovery Park. Alongside our mobile app, we will also have a web application where organizations and departments may create an account and submit their events so they will be displayed on the DiscoEvents app. DiscoEvents is not affiliated with OrgSync and neither will it hold all of its features.

.

## 2.2 | Product features

DiscoEvents major features are:

* a homepage
* an interactive calendar,
* an organizations page,
* a search bar

The interactive calendar is a swipe able and clickable calendar. Based on the given day the user clicks on, a pop up will be created that will display all of the events going on that day along with some basic information about the events.

The organizations page is a scrollable page displaying all of the discovery park organizations registered through DiscoEvents. Upon selecting an organization, the user will be taken to a page for the organization. The page will display a photo for the group, basic group information, as well as future events for the group.

The search bar searches all of the future events in Discovery Park and returns the event information to the user. The search function will work in two ways. The first is a basic search, matching with anything that is searched. The other is a search by tags, in which users will be able to select the tags that have been used to describe the future events, and results will be given based on the tags.

## 2.3 | User Classes and Characteristics

Student user class: This group will encompass the average Discovery Park student. Their use of the app will be to gain information of the events going on, as well as familiarize themselves with the application. The anticipated use for this user class is 2-4 uses of the app per week with a max of 1 minute per use. This user class will be our main class, and our app will be built primarily for this audience.

Organization user class: Departments and organizations may submit their events to DiscoEvents so their events will be available for students to see. Their use time will not be on the android app, but rather it will be on our web application where they can submit events. Their use will be to spread awareness of their events and gain attendance for their respective events. The anticipated use for this user class is 1 use a week with a max of 5 minute per usage to submit an event. This is not our main user class, however their involvement is crucial to filling the DiscoEvents app with data.

## 2.4 | Operating System

The mobile software will operate on the Android OS. It will be developed for Android versions 4.0 and above. This is because over 90% of android users run the aforementioned OS, allowing us to reach a large amount of android users. The app will require 20 megabytes of storage to be saved and installed on the phone.

The web application will operate on the web. It will be developed to work on versions of Google Chrome, Safari, and Firefox from 2008 and on. The web application will be created with basic browser functionality in mind, so that no additional plugins are required.

## 2.5 | Design and Implementation Restraints

Our biggest design and implementation constraint is our inexperience. We have decided to undertake this project in Android Studio and Django to grow our knowledge in the process of creating this app. What this means is that we may not be able to produce the finest or most polished product. However we intend to produce a strong application with all of the requirements specified in this document.

## 2.6 | User Documentation

Within the application itself, we will have a help section. The help section will cover the basic uses of the application, how to search, and how to report bugs. Along with it will be a FAQ section, and finally, an email that will be available for users to contact with any further inquiries or help needs that had not been satisfied with the built in help.

Within the web application, we will also have a help section. The help section will cover how to create an account, and how to post your event for it to be moved to the mobile DiscoEvents app. A FAQ section will also be included, alongside an email that will be available for user to contact with any further inquiries or help needs that had not been satisfied with the built in help.

# **3 | System Features**

## 3.1 |Android App Features

### 3.1.1 | Home Screen

#### 3.1.1.1 Description and Priority

The home screen will have slideshow of upcoming events and will have the navigation bar along the bottom. The home page is high priority because it is the first page a user will see.

#### 3.1.1.2 Stimulus/Response Sequences

If the user clicks the slides the app will link to the event shown. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Search: the search button sends users to the search page
  + Organizations: the organizations button sends user to the list of organizations
  + Calendar: the calendar button sends user to the calendar page

#### 3.1.1.3 Functional Requirements

The functional capabilities for the Home Page consist of linking the navigation bar buttons to their proper pages and creating a link (query) to the database to display the 10 next upcoming events in the slideshow and linking each one to its respective Event View.

### 3.1.2 | Calendar

#### 3.1.2.1 Description and Priority

The Calendar view will show a calendar for that month, with navigation to future months, and when you click on a day it will bring you to the events for that day. This feature is medium to high priority. The app could be created without it but will help with ease of navigation.

#### 3.1.2.2 Stimulus/Response Sequences

If the user clicks the month navigation the corresponding month will show on the page. If the user clicks on a single day the app will move to the list of events that date. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Home screen: the home button returns the user to the home screen
  + Search: the search button sends users to the search page
  + Organizations: the organizations button sends user to the list of organizations

#### 3.1.2.3 Functional Requirements

The functional capabilities for the Calendar consist of linking the navigation bar buttons to their proper pages and showing an interactive calendar where each day is linked to a page with the events for that day. Also there will be a navigation button to move between different months.

### 3.1.3 | Events of the Day

#### 3.1.3.1 Description and Priority

The events of the Day page is a simple list with the results of a query from a specific day. The priority of this is medium, and can be created later in the process.

#### 3.1.3.2 Stimulus/Response Sequences

If the user clicks on a result it will link them to an event page. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Home screen: the home button returns the user to the home screen
  + Search: the search button sends users to the search page
  + Organizations: the organizations button sends user to the list of organizations
  + Calendar: the calendar button sends user to the calendar page

#### 3.1.3.3 Functional Requirements

The functional capabilities for the Events of the Day consist of linking the navigation bar buttons to their proper pages and creating a link (query) to the database to display the events on that day and link the results to their respective Event Page.

### 3.1.4 | Organization List

#### 3.1.4.1 Description and Priority

The Organization List will have a list of all of the organizations that have events submitted. This is low priority and could be left out without affecting the overall ability of the application.

#### 3.1.4.2 Stimulus/Response Sequences

If the user clicks one of the organizations it will link it to the organizations profile page. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Home screen: the home button returns the user to the home screen
  + Search: the search button sends users to the search page
  + Calendar: the calendar button sends user to the calendar page

#### 3.1.4.3 Functional Requirements

The functional capabilities for the Organizations List consist of linking the navigation bar buttons to their proper pages and creating a link (query) to the database to display all the organizations that have events and linking each organizations to its respective Organization Profiles.

### 3.1.5 | Organization Profile

#### 3.1.5.1 Description and Priority

The Organization Profile will show an organizations details, including, name, contacts, department relation, and the organizations currently listed events. This is medium priority because it can be linked too from two separate places.

#### 3.1.5.2 Stimulus/Response Sequences

If the user clicks on one of the organizations listed events it will go to the event’s page. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Home screen: the home button returns the user to the home screen
  + Search: the search button sends users to the search page
  + Organizations: the organizations button sends user to the list of organizations
  + Calendar: the calendar button sends user to the calendar page

#### 3.1.5.3 Functional Requirements

The functional capabilities for the Organization profile consist of linking the navigation bar buttons to their proper pages and creating a link (query) to the database to display the organizations information, and there needs to be links from the list of the organizations events to the respective events pages.

### 3.1.6 | Search Page

#### 3.1.6.1 Description and Priority

The search page will have pull down menus that the user will use to narrow down the search for events. This feature is high priority.

#### 3.1.6.2 Stimulus/Response Sequences

The user will interact with several pull down menus, which include date, organization, department, tags, and event type. If the user clicks the search button the user will be navigated to a list of events. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Home screen: the home button returns the user to the home screen
  + Organizations: the organizations button sends user to the list of organizations
  + Calendar: the calendar button sends user to the calendar page

#### 3.1.6.3 Functional Requirements

The functional capabilities for the Search Page consist of linking the navigation bar buttons to their proper pages and creating a form for the user to fill out to narrow down a search of events. The “search” button on the page will call the query from the database and link to the Search Result page.

### 3.1.7 | Search Results

#### 3.1.7.1 Description and Priority

The Search Result page will show the results of a search, and have links to Event View pages.

#### 3.1.7.2 Stimulus/Response Sequences

If the user clicks the event link it will take the user to the Event View page for the specific event. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Home screen: the home button returns the user to the home screen
  + Search: the search button sends users to the search page
  + Organizations: the organizations button sends user to the list of organizations
  + Calendar: the calendar button sends user to the calendar page

#### 3.1.7.3 Functional Requirements

The functional capabilities for the Search Page consist of linking the navigation bar buttons to their proper pages and each search result needs to be linked to their respective Event Page.

### 3.1.7 | Event View

#### 3.1.1.1 Description and Priority

The Event View shows all the details for a specific event. This has high priority because the goal of our app is to show events to the users.

#### 3.1.1.2 Stimulus/Response Sequences

If the user clicks the organization that is holding the event they will be sent to the Organization Page. If the user clicks one of the options on the navigation bar it will bring them to one of the four following places:

* + Home screen: the home button returns the user to the home screen
  + Search: the search button sends users to the search page
  + Organizations: the organizations button sends user to the list of organizations
  + Calendar: the calendar button sends user to the calendar page

#### 3.1.1.3 Functional Requirements

The functional capabilities for the Event View consist of linking the navigation bar buttons to their proper pages and showing the information of the event from the database, and linking the organization to the Organization View.

## 3.2 | Django Web-based Application

### 3.2.1 | Login Page

#### 3.2.1.1 Description and Priority

The login will allow organization users to access the Django application using their account. This is high priority so that organizations have their own private accounts.

#### 3.2.1.2 Stimulus/Response Sequences

If the user is signing in to an account they will input their email and password. If the user is new they will click the new account button

#### 3.2.1.3 Functional Requirements

The page will need to connect to the database to confirm the entered information after the user clicks Login and link to the Organization Information Page, and the “new user” button will need to link to the New User Sign-up page.

### 3.2.2 | New User Sign-up

#### 3.2.2.1 Description and Priority

The New User Sign-up page will be where a club user can register for an account. This is high priority because there needs to be users to input events.

#### 3.2.2.2 Stimulus/Response Sequences

The users will input the data to create their account, and click a button to create account and bring them to the Organization Information Page

#### 3.2.2.3 Functional Requirements

The page needs to collect input, and then link to the Organization Information Page. This information will have to be saved to the database

### 3.2.3 | Organization Information Page

#### 3.2.3.1 Description and Priority

This page will hold all the information associated with the logged in account. The information can be edited as well. This is a high priority feature.

#### 3.2.3.2 Stimulus/Response Sequences

The data should be able to be edited by the user, and then a save button should send new data to the database. The user should be able to see a list of previously created events, they should be able to click on a previous event to link to the Event view to edit it, or they can create a new event.

#### 3.2.3.3 Functional Requirements

Access to the database to show saved information, and ability to edit the database when info is changed. There should also be links from previously created events to the event page.

### 3.2.4 | Create/Edit an Event

#### 3.2.4.1 Description and Priority

This page will be used to create a new event or edit a previously created event. This is a high priority because the app requires events to display.

#### 3.2.4.2 Stimulus/Response Sequences

The users should be able to input data and edit data shown if the event was previously created. There should be a save button to update information in the database.

#### 3.2.4.3 Functional Requirements

There needs to be access to the database to update information, and a link back to the Organizations Information Page.

# **4 | External Interface Requirements**

## 4.1 | User Interfaces

The PC app will have a log in/log out button as well as interfaces to post the event, to view the events you've posted, and to edit/delete the events. The Android app will have a menu on the bottom of every page showing home, search by tag, search by organization, and calendar buttons. Home button sends the user back to home page, search by tag lets the user choose which tag to include on the event search results, search by organization lets users to look up events by organization, and calendar shows a calendar that lets people search events by date.

## 4.2 | Hardware Interfaces

This app supports both Android and PC, the PC app will allow creation of accounts and submitting events, Android app will receive and show events that were entered by the PC app.

## 4.3 | Software Interfaces

The PC version of the app will be using any OS that can run the web browser, it identifies the user accounts and the event that is being posted, so people will know which account will be posting the new event. Event will be monitored by the admins and only the verified events will be sent to Android app. The Android app will require android to function, it has the purpose of showing the events on the app, as well as its menus and the event details and it will receive verified events from the PC app.

## 4.4 | Communications Interfaces

This app will use the communication functions of a web browser, and we will use HTTP for this app on the PC side.

# **5 | Other Nonfunctional Requirements**

## 5.1 | Performance Requirements

Performance requirement when pulling data for calendar:

The calendar view will be the resting view of this application. This means that when this view first appears it will need to pull data from the database about the events happening in the current month that it is. When the month is switch by the user, the calendar will need to display the selected months' data. This means that the software will need to be able to pull from the database quickly and retrieve the right information.

Performance requirement when pulling data for event or organization information view:

The event and organization view will only need information to be pulled from the database one time. This information will need to be pulled in real time when the user asks for it.

Performance requirement when tags are selected:

When the user selects tags to help their searching process the system will need to search for events or organizations that fit these tags and switch views. The new view will either display a list of the different events and organizations that contain any of those tags or will display an error message. The database will need to look through all of the tags to return the events and organizations that fit the tags in real time.

## 5.2 | Safety Requirements

The only harm that can come with using this application is misinformation. If misinformation is found then the event or organization will be set as inactive on the database and thus will not show up on the android app. After this an email will be sent out to the creators of this event. To safeguard from this we will need to set up an administrator access to the website. The administrators of the website will be able to review the events before they are set to activate in the database. There are no external policies or regulations that state safety issues that affect the product’s design or use or any safety certifications that must be satisfied.

## 5.3 | Security Requirements

### Database Security Requirements

The safety protocol we will need to take with this product are for the database. We will need to make secure our database from hackers. This includes making sure the queries into the database do not leave any openings for hackers to change, input, or drop data.

### User Identity Authentication

The web system will have a log in screen where the users will have to enter their log in information in order to access their information or create a new event. This information will be stored in the database and encrypted. If the log in information matches what is in the database then the user is allowed to log in.

## 5.4 | Software Quality Attributes

Some important quality characteristics for the product are correctness, maintainability, and usability. Correctness of the product is important because it if the information on our application is consistently wrong users will stop relying on our application. The correctness of our application should be that 85% of the information on our app is correct and 100% of the data pull from the database is the correct data. The software needs to be maintainable for the administrators so that the events are up to date and new events are added to the android app within 48 hours of the information being put into our database. The usability of our apps must be easy in the sense that the user must have an easy time traversing through the apps. Our applications should allow the user to find any piece of data within 5 clicks.

# **6 | Other Requirements**

## 6.1 | Database Interface

A database will be created using SQLite. This database will be accessed by the android interface and the Django application.

Part 2

Deliverable II

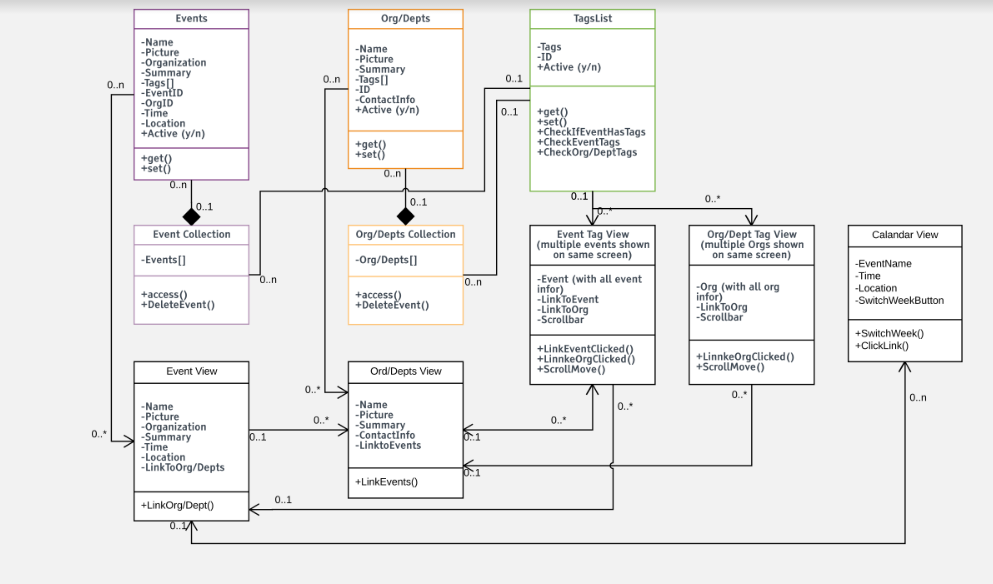
**for**

DiscoEvents

Novus

# **Part 2 | UML Documents**

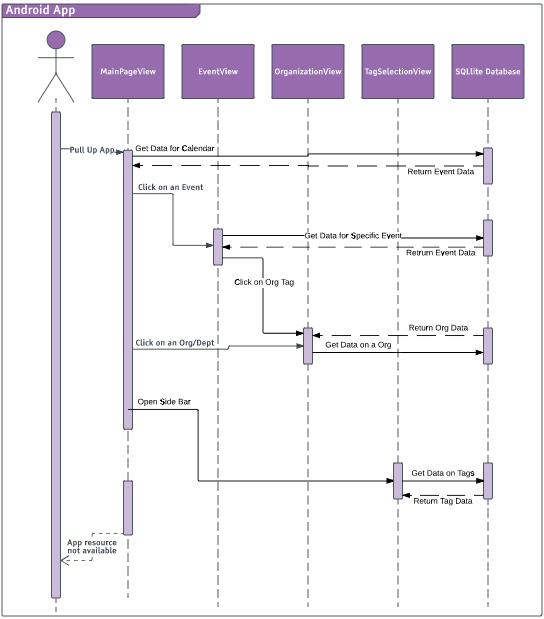
## Class Diagram Android



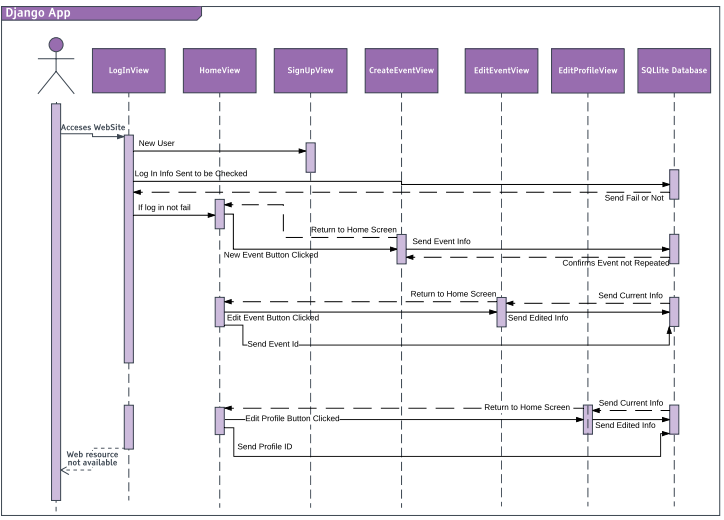
## Class Diagram Django

# 

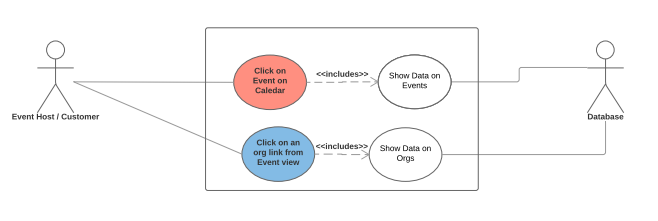
## Sequence Diagram Android



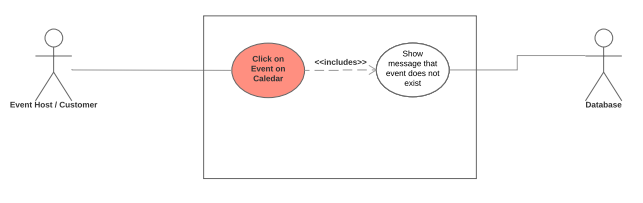
## Sequence Diagram Django



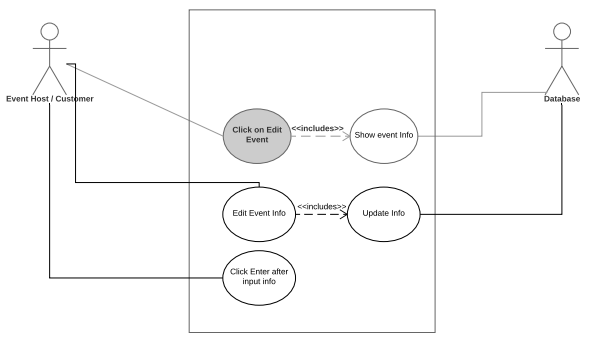
## Use Case Android



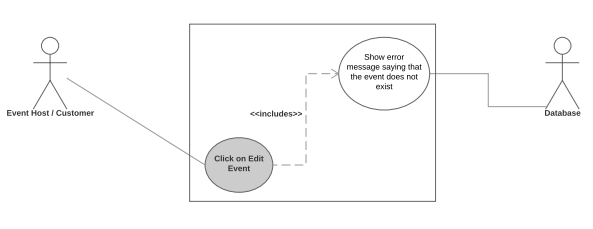
## Use Case Android Failure



## Use Case Django



## Use Case Django Failure



# **Part 3 | Test Plan**

Test case 1: PC account registration

Incorrect input: Missing fields or fields are already in database

Pass criteria: Throws an error showing which of required fields are missing or are duplicates of already existing account

Correct input: Filled fields and no fields are the same as the ones in the database as duplicates

Pass criteria: The new account's info should be created in the database

Test case 2: PC login

Incorrect input: Missing or mismatched username/password

Pass criteria: It should give out an error saying that the username/password does not match the ones in database

Correct input: Filled required field with matching username/password

Pass criteria: It should successfully login the account

Test case 3: Event creation

Incorrect input: Missing fields

Pass criteria: Throws an error showing which of required fields are missing

Correct input: Filled fields

Pass criteria: The new event's info should be created in the database

Test case 4: Event viewing

Pass criteria: The event should show up on the android app after event is published from the PC

# **Part 4 | Updated Risk Management**

## Lack of experience with Android/Django

The risk management that we developed at the beginning of the project is still applicable currently.

## Losing a member

The risk management that we developed at the beginning of the project is still applicable currently.

## Bad communication

The risk management that we developed at the beginning of the project is still applicable currently.

## Conflicting code

We will check if there are redundant or conflicting code that does not really contribute to the program itself.

If there are redundant or conflicting codes, we should comment it out and then run the program again to see if program still works, and only delete after knowing that program works without the redundant or conflicting code.

We can talk to each other to know which chunk of code does what so there won't be much more redundant or conflicting code to be made.

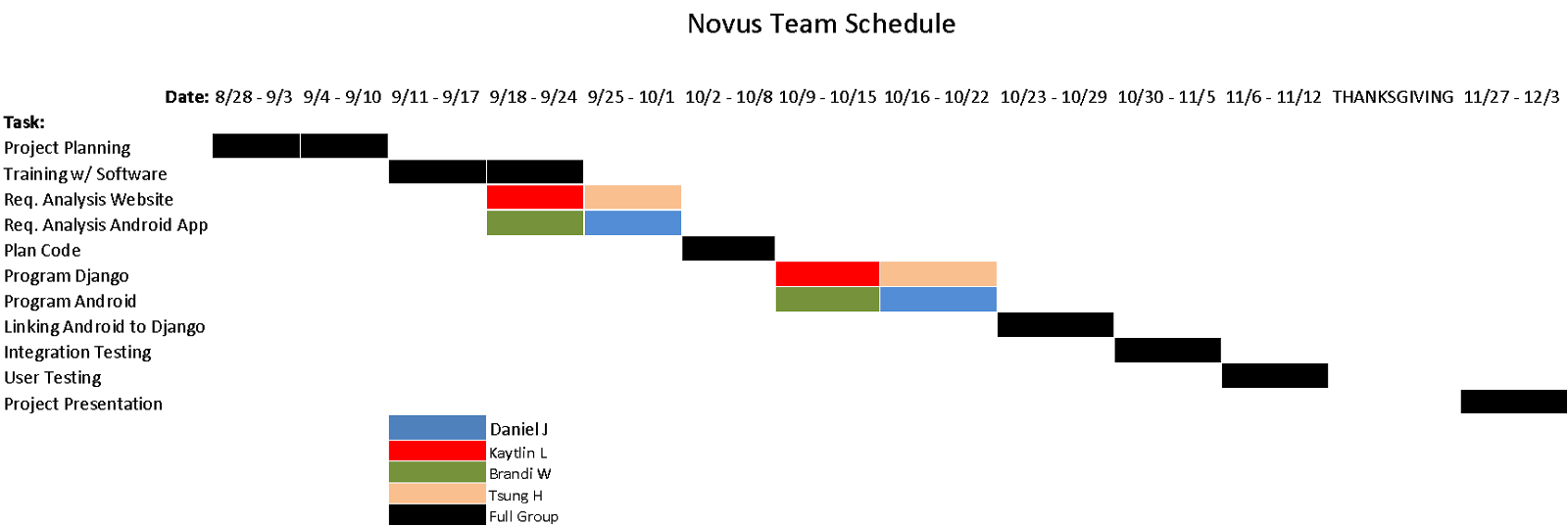
## Planning problems

We can check if there are any loopholes or exploits on our project plans and if any, we can patch it up.

If there are any loopholes, exploits, or unforeseen problems on the plans, we can adjust the plan to get rid of the problems or we can start a new plan together using the old plan as guideline.

We will check the plan more thoroughly so there will be less of a risk of unexpected problems.

# **Part 5 | Updated Project Plan**



The development schedule for DiscoEvents has been on track so far. We saw no need to make any modifications from the schedule we had used on deliverable 1. Assuming any issue arises, changes to the schedule are still viable.

# **Part 6 | Meeting Minutes**

## 9/21/17

Location: Class

Present:

Brandi Werner

Kaytlin Lafleur

Daniel Jimenez

Tsung-Han Hsieh

Agenda:

Elicitation W.S.

Requirement analysis

Deliverable 2 tasks

Progress on deliverable 3

Elicitation W.S. - we received extremely helpful insight into our application from our elicitors. They outlined certain aspects of this calendar they believed to be important based on their role as either a common engineering student or an organization leader. They stressed the importance of

> App notifications

> Individual tags for search

> Inclusion of an RSVP form on the web app

We received a lot of other information that has the possibility of being included in our project, however it may not meet the scope we are aiming for. Some of these are user profiles, event reminders, and more social media-like aspects. While we did like some of these ideas, it is not under the type of application we had planned to make.

Requirement analysis - After reviewing the elicitators information, we did a quick brainstorm of the minimum baseline requirements we would like to work towards in our app such as

Deliverable 2 tasks - We reviewed the requirements for deliverable 2 and have split them accordingly. Within the first section (which we anticipate to be the largest based on the template given in class), we split that task into subtasks among us as well. Although early, we did this to have a general idea of our roles and in case anybody had the chance to work ahead.

Progress on deliverable 3 - as a quick touch-and-go conversation, we made sure everybody was researching into their respective part for the creation of the actual application.

## 9/26/17

Location: class

Present:

Brandi Werner

Kaytlin Lafleur

Daniel Jimenez

Tsung-Han Hsieh

Agenda:

Increased Github use

Django team

Layout of app

Deliverable 2 planning/progress

Review timeline

Increased GitHub Use- we would like to shift our focus away from google docs use and focus on GitHub in order to become accustomed once we move to GitHub for the coding portion of the project

Django Team - touched base with the Django team to determine progress on learning. Progress is currently not very advanced, however there is still time to learn.

Layout of app - as a group, we designed and drew out how our android application will look. It will have 3 pages - a calendar page, an organizations page, and a search page. There will be a static menu bar at the bottom. A home icon will also be included, which will lead to a page showing all events going on today. Layout of Django app is not of critical importance.

Deliverable 2 planning/progress - We reiterated our roles and responsibilities for this second deliverable. Everything was in line, and writing for the deliverable will commence after class as decided by group.

Review timeline - while taking a close look at the timeline, we have determined that as of so far we are on track. However, we may make slight modifications and expansions to the schedule in order to further guide us in completing the project.

## 9/28/17

Location: class

Present:

Brandi Werner

Kaytlin Lafleur

Daniel Jimenez

Tsung-Han Hsieh

Agenda:

Deliverable 2 Section 1

Further Deliverable 3 planning

Deliverable 2 Section 1: based on the template provided by our instructor, we reviewed the most in depth section of deliverable two, which is section one. Based upon the roles we felt most comfortable with, we split section one into four parts amongst our members. Brandi completing template part 1, Daniel completing template part 2, Kaytlin completing template part 3, and Tsung-Han completing template part 4.

Further Deliverable 3 planning: Brandi and Kaytlin put forth their respective ideas on the design and usability aspects of our app. As a group we decided what aspects to implement. One of these is many links between pages. For example from the calendar view you could see an event by IEEE. From there you could click on the event and it will take you to the IEEE organization page. Other details are minuscule design choices.

## 10/5/17

Location: class

Present:

Brandi Werner

Kaytlin Lafleur

Daniel Jimenez

Tsung-Han Hsieh

Agenda:

Deliverable 2 tips from teacher

Deliverable 2 touch and go

Deliverable 2 tips from teacher: our class lesson today was focused on deliverable 2.

Tips we gained are:

* Be precise, clear and unambiguous
* Write readable, complete sentences
* Write in specific words, as opposed to vague

Deliverable 2 touch and go: as a quick group meeting, we discussed our own progress on the deliverable. By this point we are nearly finished, and have to focus more on minute details as well as document formatting. No group member is behind on their role.

# **Part 7 | Progress Report**

The progress of our project is going well. The reason we say this is because we are right on track with the timeline we made on deliverable 1. Just last week we finish researching and studying both android and Django requirements. This upcoming and the following week will get the base and core functionality of our software programed.

All our team members have a good grasp on what they need to be working on and the pace they need to get their task done. We have accomplished this through good communication and having meetings twice a week. The Android programmers are working closely together and have broken their task up between themselves. One of the Android developers will be working on the calendar view while the other will focus on the searching aspect of the program including tags. The Django programmers will be working together on all aspects of the web application front. After the next two weeks we will be able to start linking the two development platforms together through SQLite and start user testing.

At this point in time we are all feeling comfortable with the pace we have given ourselves and feel that we will be able to accomplish this program in a reasonable amount of time. The only struggle that we might have in the future is if one of our development tools has a limitation that our developers cannot overcome or solve. If this happens we have already discussed about different alternatives of what to do and how to solve these problems. One of the last resorts is to change development tools.

# **Part 8 | Member Contribution**

|  |  |  |  |
| --- | --- | --- | --- |
| Member Name | Contribution Description | Overall Contribution (%) | Note ( if applicable) |
| Brandi Warner | Part1.1.\*, Part 1.5.\*, Part 2 UML Design Documents, Part 7 Progress Report | 28% |  |
| Daniel Jimenez | Part 1.2.\*, Part 5 Updated Project Plan, Part 6 Meeting Minutes | 24% |  |
| Kaytlin Lafleur | Part 1.3.\*, Part 1.6 , Part 8 Member Contribution Table, Assembly of Final Document | 24% |  |
| Tsung-han Hsieh | Part 1.4.\*, Part 3 Test Plan, Part 4 Updated Risk Management | 24% |  |