**PROJECT GLOBUS**

COSC 4730

Final Project

December 3, 2014

Taylor Olson

Jesse Miller

Kelsey Crea

**ABSTRACT**

Project Globus is a group management system designed to work for a variety of groups, both casual and professional. Whether it’s a group of friends planning a weekly board gaming session, student’s working on a school project or a business team working on a company’s next biggest product, project Globus will provide an innovative, simplistic and versatile experience to keep the group organized via document sharing, group notifications, group calendars and group messaging.

1. **Introduction**  
   Project Globus is our group project that is a group management system for our Senior Design course. We decided to implement the ideas for the Android portion of Project Globus for our COSC 4730 final project. This would give us the opportunity to learn and work with the Android app before we implement all of the cross platform compatibility. Throughout this paper, we talk about how the project started and what we did to plan, the design of the user interface and what went into that, as well as how the primary features of the app are developed. Finally, we will conclude with a major error hang-up that we ran into. Project Globus is a group management system designed to work for a variety of groups, both casual and professional.
2. **Origins/Preplanning/Planning**  
   Project Globus was conceived in the back of a 1986 Buick in the alleys of Historic Downtown Laramie. This idea started from the needs of Taylor’s business fraternity, Alpha Kappa Psi, for better group collaboration software. The team decided to stem this project into COSC 4730, Mobile Programming, as well as Senior Design. Jesse had an epic epiphany and decided the name would be “Project Globus.” The team also includes David Crane, who is working on the back-end server-side database work that is not relevant to the COSC 4730 portion of this project.   
   As a team, we sat down and wanted to strategically plan the project before hitting Eclipse. We sat down and figured out what features would be in the scope for our project, and what would be too much. The primary features we decided on in that meeting were document sharing, group notifications, group calendars, and group messaging.   
   We set our schedule for both Senior Design and COSC 4730. Our schedule for COSC 4730 was as follows:

* 10/22 – Finish Proposal
* 10/29 – Have Database Running and Functional w/ Tentative Tables
* 11/1 – Initial Setup Functional and Feeding to Database
* 11/8 – Feed functional, calendar functional
* 11/15 – Integrate system together
* 11/22 – Have Project Completed and Functional
* 11/28 – Have beta tested and debugged/Create Presentation/Paper

While that schedule didn’t get followed completely, we tried to keep that as a decent timeline to keep us in check and on time for our project. Before we even started on the development project, we developed a tentative flow chart, which Kelsey built on and expanded.

1. **Flow Chart**

As previously mentioned in the planning process, we discussed the flow of our program in a planning meeting. As a group, we determined that a waterfall design methodology was appropriate for our application because we wanted to tackle the project starting from the login screen to the exit screen in terms of functionality. In line with the waterfall approach, we decided that the program would be split into two different activities; one to handle the login pages and one to handle the group selection and calendar management. We split the application into two different activities because once a user has logged into the Globus app, he/she will not need/want to navigate back to the login page often. This is when we began creating our flow chart. The flow of the application starts from a welcoming login page, and moves towards a group selection page, which then loads the main group interaction page including a message feed and access to google drive and a master group calendar.

1. **User Interface**

User Interface (UI) design was something that as a group, we collectively agreed, needed to be simple, elegant, and straightforward. Our target audience for this app encompasses a large demographic, so we cannot assume any sort of technical experience. With this, we took a very simplistic approach to our UI design. We decided on a simple gray interface with one color accent. After much debate, we settled on blue. This would make it so the user could focus on what’s important, the content. Font style within our app is a simple sans serif font that is easy to read and not distracting.  
Because the “Whiteboard” would be the primary interface that the user would interact with, this design was also a collective effort with the group. We decided on using a rough “25/75” vertical split of the screen. Our app is currently vertical orientation locked, and we will use 25 percent of the width for large navigation buttons to different features of the app, and the other 75 percent of the app would be the “Whiteboard.”

Our primary focus of the design for project Globus is to give the user an easy, quick access interface.

1. **Login/Group Creation**  
   The first activity has a main welcome fragment to the Globus application where the user can either login or create a new Globus account. If the login button is selected then a new fragment is loaded which lets the user enter both a username and password. This information is then checked against the database and, if valid, the first activity ends and the user name is passed to the second activity. If the user selected the “create a new account” button, then a different fragment would have loaded that lets the user enter in an email and password. If the email and password are valid, the first activity will start, and the username is passed from the first to second activity. When the second activity has started a fragment will load that displays the groups that the user is already a member of in addition to two buttons that let the user either create a new group or join an existing group. If the user selects the join a group button then a new fragment will load that lets the user enter in a groupID number and group password. If these credentials are valid when checked against the Globus database, the user will now become a member of the group and the main group page will load. If the user were to select the “create a new group” button then a new fragment will load that lets the user enter a group name and group password. Again, if these credentials are valid when checked against the database, the main group page will load. The user also has the option to click on the names of the groups that they are already a member of in order to load the main group page. Once the main group fragment page has loaded, the user will see two different fragments sharing the same screen. One fragment, consisting only of buttons is displayed down the left hand side of the screen, and the other fragment is what we have deemed ‘The Whiteboard’, which is a message feed, displayed on the right hand side of the screen. The group name is displayed at the top to inform the user of which group page they are viewing. The user also has the option to navigate back to the group selection page in order to select a different group page that they would like to view.
2. **Whiteboard**The Whiteboard is essentially the “main hub” of the Globus group interface. This is where all updates within the group appear, whether they are notifications for upcoming events, updates talking about new documents uploaded to the groups’ Google Drive account, or simply messages telling everyone to have a good weekend. The Whiteboard also houses buttons that will take the user to all other sections of the group interface such as the calendar and google drive sections. In order to implement this, we created a fragment that contains two fragment views inside of it. The fragment view that contains the buttons was placed on the left side of the screen, while the fragment view for the Whiteboard (and other group related sections) was placed on the right. This was done in order to make the UI more consistent and to make it appear more simplistic by making the buttons, and their position on the screen, independent of the different group sections.

In order to populate the Whiteboard with events and messages, we pull information from two different fields in our database. These fields are the events and messages fields. When the system queries these fields, it compiles the results from both, sorts them by date (date created for messages, and the start date for events) and then populates the Whiteboard with the results, displaying the closest upcoming events and most recent messages first. While the Whiteboard notifications only originate from two different database fields, it’s important to note that these two fields encompass a wide variety of situations. The Calendar events will cover both company related events, and any other events worth keeping track of such as birthdays, deadlines, etc. The messages will contain things such as motivational quotes written to boost morale, updates from group members pertaining to group, and personal, projects, and new additions to the group Google Drive account.

1. **Calendar/Events**  
   Each group created in Globus has its own calendar and events associated with it. These are kept track of in the Globus database (inside of the Calendar and Events fields). In order for a user to view, and interact with, this calendar, we created the group calendar view fragment. This fragment contains an android calendar view, the name of the group, a back button, and an “add event” button. The calendar view object displays information queried from a local calendar created on the user’s android device. These calendars are created via androids calendarContract object (an object used to interface with android’s calendar data model). The first time the user opens up a group’s calendar, a new calendar will be created on the device. This calendar will then be populated by querying the events field in the Globus database and inserting them into the local calendar.

In order to add an event, the user will press the “add event” button. This button opens up the new event fragment. This fragment contains input fields for all the information required to create the event (start/stop time, name, description, etc.). Once the user officially creates the event, it will be sent to both the local calendar and to the Globus database.

1. **Google Drive**  
   At its base the Google Drive view consists of a webpage linking directly to Google drive, and two text fields that display the groups account name and password. From here, the user will be able to log in to Google drive, and access any and all of the groups documents. This is one of our simplest views/sections.
2. **Database Issues**  
   For our Senior Design project, Globus is going to transform into a cross-platform application that will be used with Android, iOS, and a web interface. This will require a server-side database and the functionality to be retrieved by the phones, and the database would be independent of the phone. We wanted to do a smaller version of that with our COSC 4730 project, but we ran into some major errors and hang-ups. We initially tried by working with Jim Ward on getting some server space available for us to use. Once we got this up, we ran into issues where the database was only accessible from within the Computer Science network. We decided to move to Amazon servers, which would dismiss our issues of being on a particular network. David has been the Senior Design team member to work on all the server-side database code, and he wrote an excellent Java client that we would be able to use with Android to interface with the database on the server.

Attempting to implement this is where we ran into problems. When we tried implementing David’s database code with our application, it would cause the app to crash. After many hours of trial and error, and a conversation with Jim Ward, we tried to use a MySQL connector that was compiled for Android as apposed to one that was compiled just for Java. This solved most of our issues, the only issue we were running into was a connection issue from David’s code. After checking to make sure we had the Android Internet permission set, we did a lot of research to find that Android would not allow for the open connection via the port to the server. Many people across the development community have run into the same issue.

At this point, we decided to make Globus a local app for COSC 4730. This means that when a group is created, it is then saved to a Globus database on the phone it is created on.

1. **Conclusion**Project Globus is a group management system designed to work for a variety of groups, both casual and professional. We started out by planning the project and creating a schedule, making sure we were going to cover all of the aspects that we needed to. Then we went to designing the UI and the logic behind it. Finally, we were faced with a major challenge of using a remote database with the phones, and we needed to come up with a solution for that. Overall, this project has been a learning experience that will help us finish the Project Globus system for our Senior Design course.

Attachments: Project Globus Proposal