

Our goals this week were to create the layouts for our android application, and to create a database schema for managing information on the back end. We were also to include user input validation in layouts containing user input forms.

The android layouts refer to the XML used to construct the visual aspect of the application. This means that there is a visualization of each screen available through the IDE layout renderer. We will be able to display these layouts during our meeting using a layout rendered in an android IDE. Functionality is not a requirement of the layouts. Therefore, buttons, text boxes, and other elements of the page may not function until they are properly implemented. In addition, the Google Maps page has been created, appropriate API keys have been generated, and permissions have been properly set up in the android manifest. We will not be able to demonstrate the Maps page until we have implemented the code backing it, however. Pages containing tabs/list views also require code to properly create them. The layouts for these pages have been created, but can not be properly seen through an IDE layout renderer. All user strings have been properly placed in the strings.xml file.

We have experimented with a traditional android holo theme, and a catamount colors theme. We have not settled on a specific color scheme as of yet, but it is trivial to alter the color scheme and will not take away from the rest of the project. There is a colors.xml file containing a multitude of colors options that we can easily search/replace project wide in order to test new color formats.

Input validation is complete from what is possible via XML. For example, we are able to create an email input box and use the format validation, but validating a valid or non-existing email must be done through code. This will be implemented later.

The database schema has been created using a program called dbschema. This program allows the user to create an appropriate ER diagram for their database. This schema can be seen from the attached image. The schema contains the tables that contain the data that must be stored server side for cases where the server must make a decision based on that data, or when other users have access to that data. For example, the server may need to decide if a user is able to receive an event invitation based on the user's privacy settings, so this information needs to be stored on the database. The database schema is normalized, meaning that information about users has been spread out properly across tables to prevent data redundancy in the database. Primary keys can be identified by a key marker on the table. Foreign keys can be identified by a magnifying glass icon. Foreign keys also suggest a relationship between two tables, which can be seen by the appropriate arrows. Arrows point from the referencing table to the table being referenced.