Rowan Chat Room Design Document

Solution Design Overview

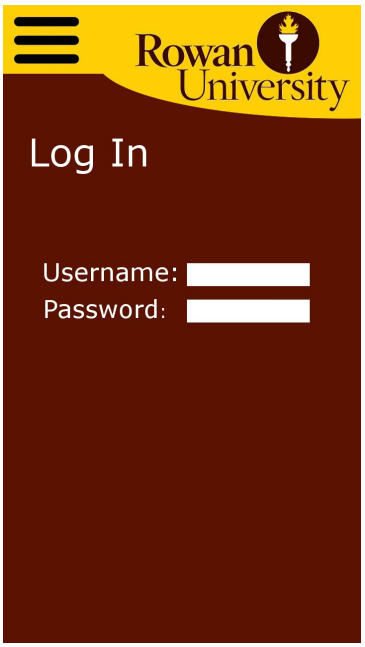
The solution for the system is two-fold: the interface and the server. The user interface is an Android app which affords the user the ability to log in, see screens which allow the user to talk to other users about different topics presented by the room. The server will be running the basic LAMP setup and will create a tunnel to the user once they try to connect, then login. The application will send data to the server including the user's message, and any additional data needed.

Technical Architecture

The hardware for the client system is any Android mobile device running Android 4.0 or later and any dedicated server either hosted through a 3rd party or in house. To use this system, all users will need a Rowan account.

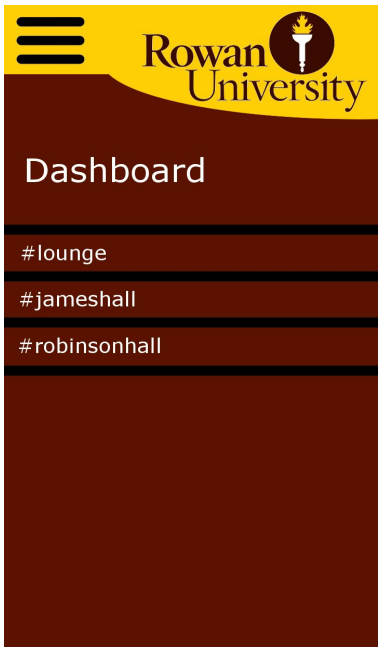
Module Description

The first module, the login screen, is a simple view with two fields for inputs: a Rowan email and password.

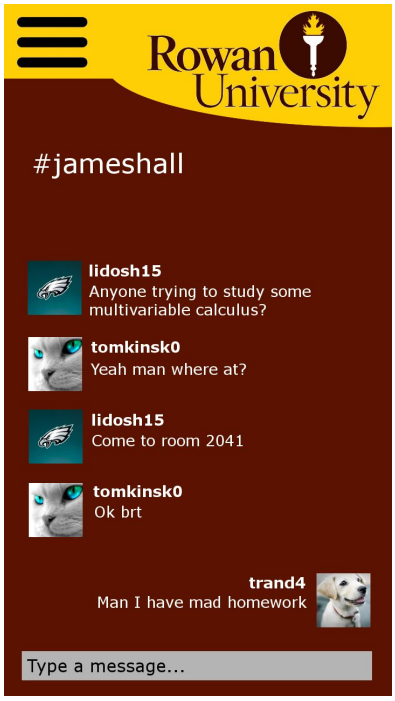


The second module, the dashboard, has various widgets which allow the user to connect to different chat room and edit information. The "Dashboard" page will show users the different chat rooms to choose from in the system, which are hosted somewhere, and acknowledged by the main server. There is a hierarchy of different chat rooms which are associated with different topics. For example, each college

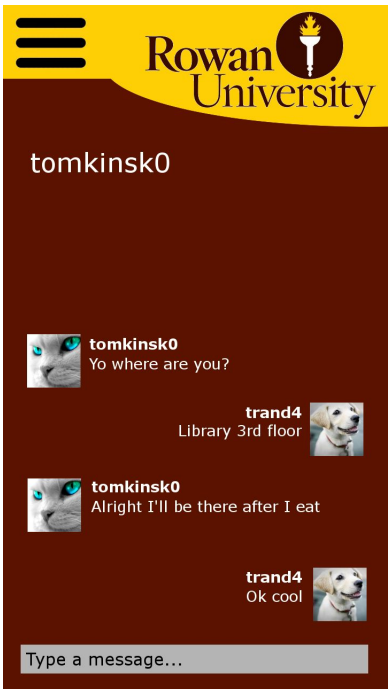
may have a chat room for discussion, but there may be topics for majors in the college, and further for classes in a major.



The “Chatroom” widget allows users to communicate with one another through updated messages on a single screen. When accessing the chat room each phone is connected to a server and through that server all messages are sent to each phone connected.



The “Direct Messaging” widget allows users to send messages directly to a user of their choice. When sending direct messages the user can either enter the user name of the receiver or click on the user name of a user in the chatroom and send them a direct message.



Backend Info

The server, to be hosted on or off Rowan’s campus will be enabled with an XMPP server, which will accept messages from clients, and relay them to the other clients. Special messages will be used for handling making changes to a user’s account, that will then need to be relayed to the database.

A database will have to be created to be implemented with the server. The database will be able to store the login information of users. The client app will send a designated string to the server to validate a change to the user’s profile. Profile pictures will be expected after this message, and sent via FTP to be cataloged by the server.

Problem Solving Solutions

Problem	Solution
How do we deal with overloading communication lines with too many concurrent users?	Limit amount of users able to connect to server to 30 at a time.
How do we deal with users that refuse to abide by community guidelines?	Create guidelines that are easily accessible and disable accounts of users that break those guidelines or temporarily put those accounts on a temporary ban, until 3 misconducts are achieved and a ban will happen.
How do we ensure users of the system are part of the Rowan community?	Have users log in to the system with their rowan credentials. This ensures that all users are either students or faculty.

Mid-Assessment

Goals to be accomplished:

- Mockup Screens created
- Chat System
- Login System
- Profile System

Tasks assigned to group members:
Harold Hatch: Developer, chat room widget implementation
Edward Jernigan: Dashboard Widget Implementation
Harrison Lidoshore: Server setup/backend
Thomas Smyth: Scrum master, interface design
Kevin Tomkins: Dashboard screen, some backend development
Darwin Tran: Edit Profile Widget Implementation

Prototype

Demonstration

- The ability to send and receive messages.
- An easy to use/learn interface
- Event driven procedures
- How each page connects to the next
- Testing procedures

Concerns

The goal of this presentation is to address concerns with the ability to connect to the server and view messages. Demonstrate how all requirements will be accomplished, and give a general view of the expectations of a finished product.