**Bulletin Board: Announceit**

**Yosiris Andújar**

**Christian Lorenzo**

**Armando Ortiz**

[Yosiris.andujar@upr.edu](mailto:Yosiris.andujar@upr.edu), [Christian.lorenzo1@upr.edu](mailto:Christian.lorenzo1@upr.edu), [Armando.ortiz1@upr.edu](mailto:Armando.ortiz1@upr.edu)

1. **Introduction**

College campus have a lot of life going on and many opportunities to take. People and organizations want to announce their events, used books, tutoring, housing and many other things. They mostly use physical bulletin boards to reach the campus community. This require much effort such as manual labor, printing papers, time, limited space and accessibility.

The target clients are students, professors, and faculty members in the campus of UPRM.

This application will be both a mobile and web-app because it will reach a broader clientele without difficulty.

The technologies that will be used are HTML5, CSS3, Javascript, AngularJs, Ionic, PostgreSQL, and Java-Play. This will result in a consistent development with both mobile and desktop applications.

1. **Client App Description**
2. **Home page**: The app will show all announcements regardless of not having a user account.
3. **Searching**:  Any user can search any post by keyword and category.
4. **Sign-up**: Users will register giving their email and personal information.
5. **Creating an announcement**: A registered user will be able to post his announcement. The user must choose a category of the announcement by books, tutoring, events, housing, or others. The description of the announcement will be in a textbox with the option of an attachment, like pictures, pdf etc.

Note: Registered users can only have one announcement in the home page. Premium users can have unlimited announcements in the home page.

1. **Messaging**: Registered user can privately message with the owner of the announcement.
2. **Payment**: Registered user can pay the owner of the announcement according to their own arrangements.
3. **Settings**: The registered user can edit their attributes.
4. **Profile**: The registered user can see their history of payments, messages, and posts.
5. **Reports**: The registered user can report a post if the content is inadequate.

Technologies. HTML5, CSS3, Javascript, AngularJs, and Ionic

1. **Server side description**

Server Capabilities

* The system shall allow the user to sign in into an existing account or create a new account.
* The signup process will send validation email after registering an account using SendGrid.
* The administrator will have access to all user data.
* The administrator will be able to delete any user account that does not abide by the site’s terms and regulations.
* User registration is not necessary for browsing the main page.
* The system will prevent spamming of announcements.
* Payment will be supported by citrus.js library
* A notification should be displayed when there is no internet connection.
* A notification should be displayed when announcements have been correctly submitted.
* A notification should be displayed when there is an error while submitting the announcement.

The server side tier is going to be developed using Java-Play, SendGrid and PostgreSQL.

|  |
| --- |
|  |

1. **E-R Diagram**



The E-R Diagram represents the tables.

The relationship between “Member” and “Message” is N by M. Multiple “Member” can exchange multiple “Messages”.

The relationship between “Payment” and “creditcard” is N by 1. Multiple “Payment” are paid with one “credit card”

The relationship between “Member” and “Announcement” is 1 by N. A “Person” posts multiple “Announcements”.

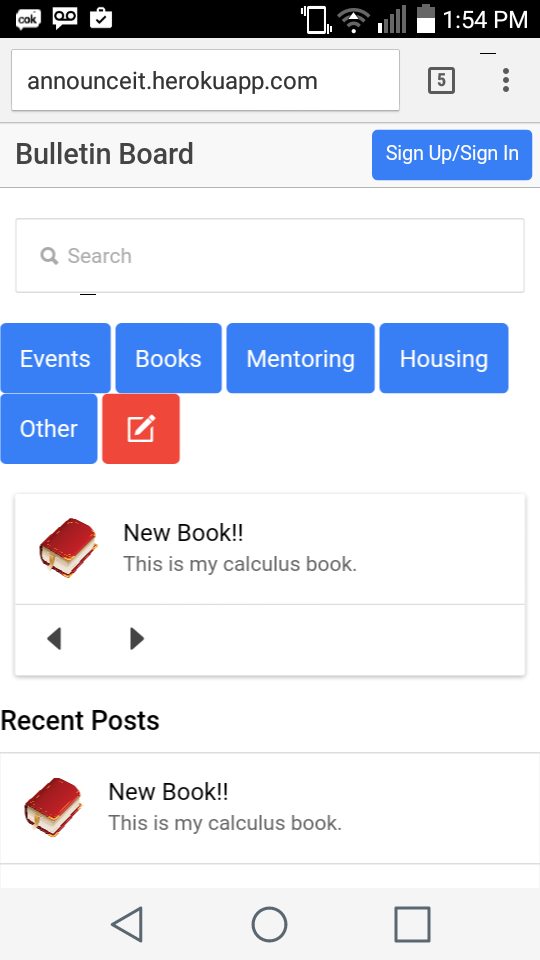
The relationship between “Member” and “Report” is N by M. Multiple “Person” reports an “Announcement”. A “Person” reports multiple “Announcement”.

The relationship between “Admin” and “Admod” is 1 by N. The “Admin” adds multiple “Admod” (Admin Modifications).

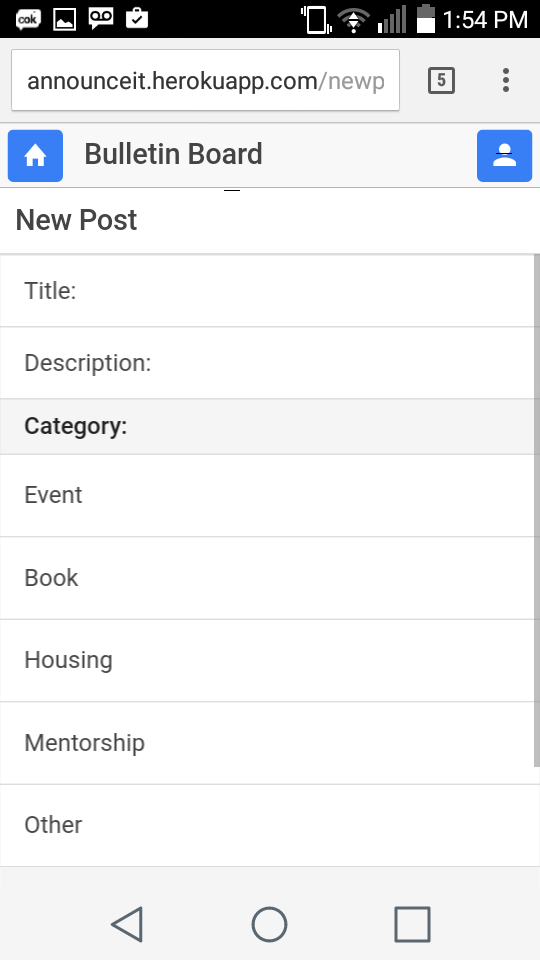
The relationship between “Member” and “creditcard” is 1 by N. The “Member” uses multiple “credit cards”.

The entities “Event”, “Book”, “Housing”, “Mentorship”, and “Other” are sub-entities of the “Announcement” entity.

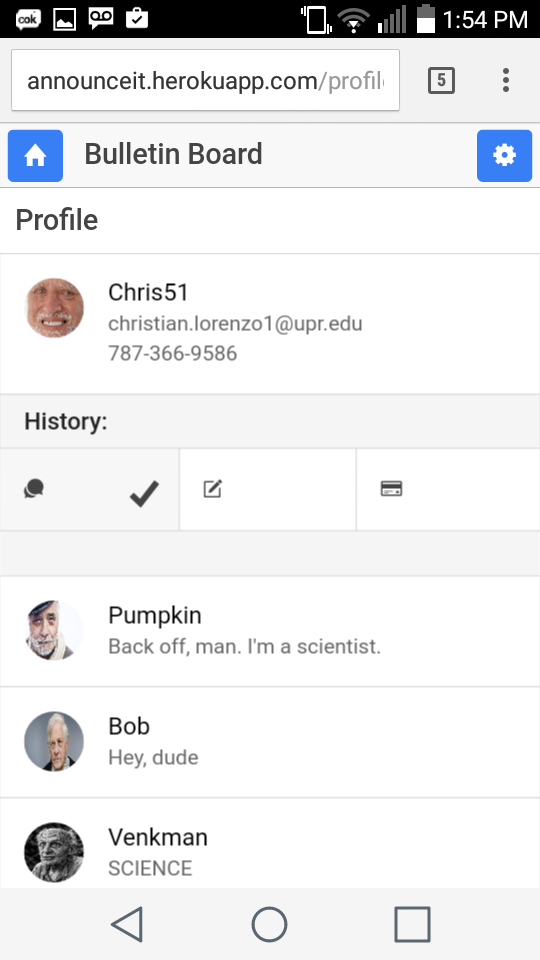
1. **Client – Side Screen Shots**



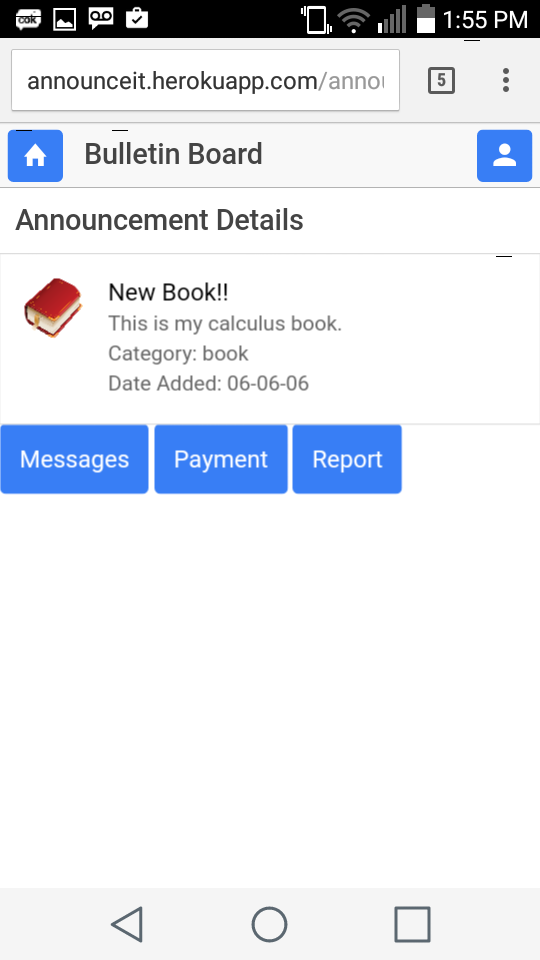
* This is the homepage for Bulletin Board.
* The “Sign Up/Sign In” button takes the user to a form where the user has to input all the necessary information to create an account or to log in into the system.
* The search will filter the results by their title or description, but is not enabled yet.
* The “Events”, “Book”, “Mentoring”, “Housing”, “Other” buttons, will filter the results by category. For example, clicking on the “Books” button will result in the view of posts that fall under the books category only.
* The button with the “write symbol” will take the user to the New Post page.



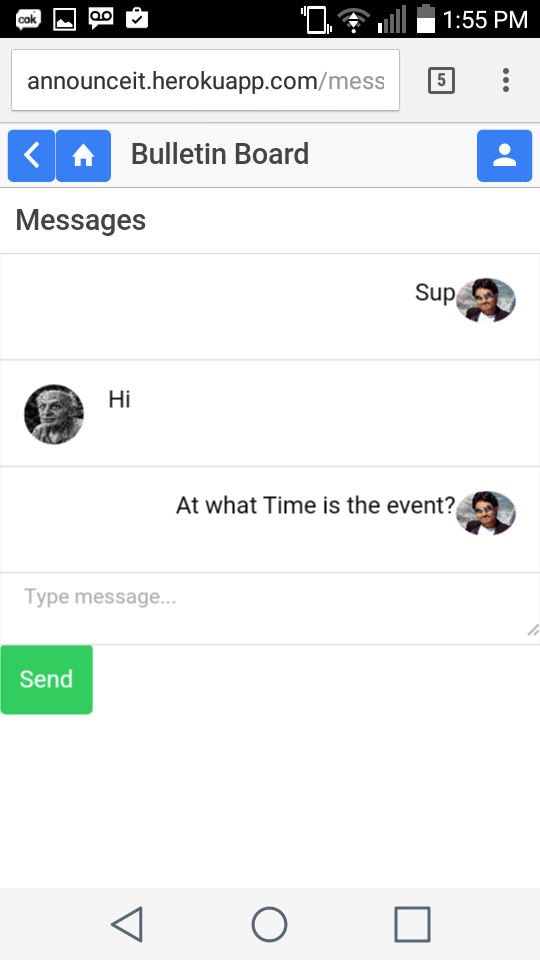
* This is the new post page.
* The user will input all the necessary information to make a new post.
* The user will select the category of the post.
* The button with the “avatar symbol” will take the user to their profile.



* This is the profile page.
* The button with the “home symbol” will take the user to the homepage.
* The button with the “gear symbol” will take the user to the settings page, where the user can edit their information.
* The profile details shown in this page will be the profile picture, username, email, and phone number. The user can select the “message symbol” to view their messages; Or the user can select the “write symbol” to view their previous posts; Or the user can select the “card symbol” to view their payment history.



* This is the announcement details page.
* The user will see the details of their posts.
* The “Messages” button will take the user to the chat page of the original poster and the user.
* The “Payment” button will take the user to a page in which they can make a payment to the original poster.
* The “Report” button will take the user to a page where they can report the post.



* This is the messages page.
* The user can send and receive messages in this page.

SQL Tables:

create table Member(

uID bigserial primary key not null,

username varchar(16) unique not null,

password varchar(20) not null,

email varchar(50) unique not null,

profPic bytea,

typeOfAccount varchar(16) default 'Regular',

activePosts int default 0

check ( (activePosts <= 3 and typeOfAccount = 'Regular')

or (typeOfAccount = 'Premium')),

phoneNumber text[]

);

create table Admin (

uID bigserial primary key references Member(uID) not null,

securityKey varchar(16) not null,

firstName text not null,

lastName text not null

) ;

create table Book(

category char(1) default 'b' not null,

postID bigserial not null,

uID bigserial references Member(uID) not null,

title varchar(20) not null,

description text,

dateAdded timestamp default current\_timestamp not null,

attachment bytea,

name text not null,

author text not null,

edition varchar(10),

year int,

price numeric(9,2) default 0.00,

primary key(category,postID)

) ;

create table Mentorship(

category char(1) default 'm' not null,

postID bigserial not null,

uID bigserial references Member(uID) not null,

title varchar(20) not null,

description text,

dateAdded timestamp default current\_timestamp not null,

attachment bytea,

subject varchar(20) not null,

fee numeric(9,2) default 0.00,

primary key(category,postID)

) ;

create table Event(

category char(1) default 'e' not null,

postID bigserial not null,

uID bigserial references Member(uID) not null,

title varchar(20) not null,

description text,

dateAdded timestamp default current\_timestamp not null,

attachment bytea,

dateOfEvent timestamp,

location text,

fee numeric(9,2) default 0.00,

primary key(category,postID)

) ;

create table Housing(

category char(1) default 'h' not null,

postID bigserial not null,

uID bigserial references Member(uID) not null,

title varchar(20) not null,

description text,

dateAdded timestamp default current\_timestamp not null,

attachment bytea,

address text,

monthlyPrice numeric(9,2) default 0.00,

primary key(category,postID) ) ;

create table Other(

category char(1) default 'o' not null,

postID bigserial not null,

uID bigserial references Member(uID) not null,

title varchar(20) not null,

description text,

dateAdded timestamp default current\_timestamp not null,

attachment bytea,

itemName text not null,

primary key(category,postID)

) ;

create table Admod(

category char(1) default 'a' not null,

postID bigserial not null,

uID bigserial references Member(uID) not null,

title varchar(20) not null,

description text,

dateAdded timestamp default current\_timestamp not null,

attachment bytea,

typeOfMod text check(typeOfMod in('Error','Warning',

'Attention','Announcement','Modification','Update')),

typeOfUsers text check(typeOfUsers in ('Regular','Premium',

'All')),

primary key(category,postID)

) ;

create table Message(

mID bigserial not null check(mID > 0),

chatid bigint not null default 1,

senderID bigserial references member(uID) not null,

receiverID bigserial references member(uID) not null,

messageText text not null,

seen text not null default 'Not Seen'

check (seen in ('Not Seen','Seen')),

dateSent timestamp default current\_timestamp not null,

primary key(mID,chatid,senderID,receiverID)

);

create table Payment(

pID bigserial not null,

buyerID bigserial references member(uID) not null,

sellerID bigserial references member(uID) not null,

cardID bigserial references creditcard(cardid) not null,

category char(1) not null,

postID bigserial not null,

amount numeric(9,2) not null,

dateOfPayment timestamp default current\_timestamp not null,

primary key(pID, buyerID, sellerID)

);

create table Report(

category char(1) default 'o' not null,

postID bigint not null,

uID bigserial references member(uID) not null,

typeOfReport text

check(typeOfReport in ('Inappropriate','Spam',

'Offensive','Scam','Other')),

comment text not null,

primary key (uID,postID,category)

);

create table creditcard (

cardid bigserial unique not null,

uid bigserial references member(uid) not null,

cardtype text check(cardtype in ('Paypal','MasterCard','Visa','Discover','American Express')) ,

cardnumber text,

primary key (cardid,uid,cardtype))