

I HEART CORVALLIS MOBILE APPLICATION: WINTER 2018 PROGRESS

Created By:
Bradley Imai
Omeed Habibelahian
Dylan Tomlinson

OVERVIEW

- “I Heart Corvallis” is a mobile application we are creating for the Corvallis Community Relations (CCR) office
- Part of the larger I Heart Corvallis initiative ran by the CCR office to encourage students to feel more connected to the greater Corvallis community
- App focuses on increasing student awareness of events happening around town and creating an incentive for students to attend these events

OVERVIEW (CONT'D...)

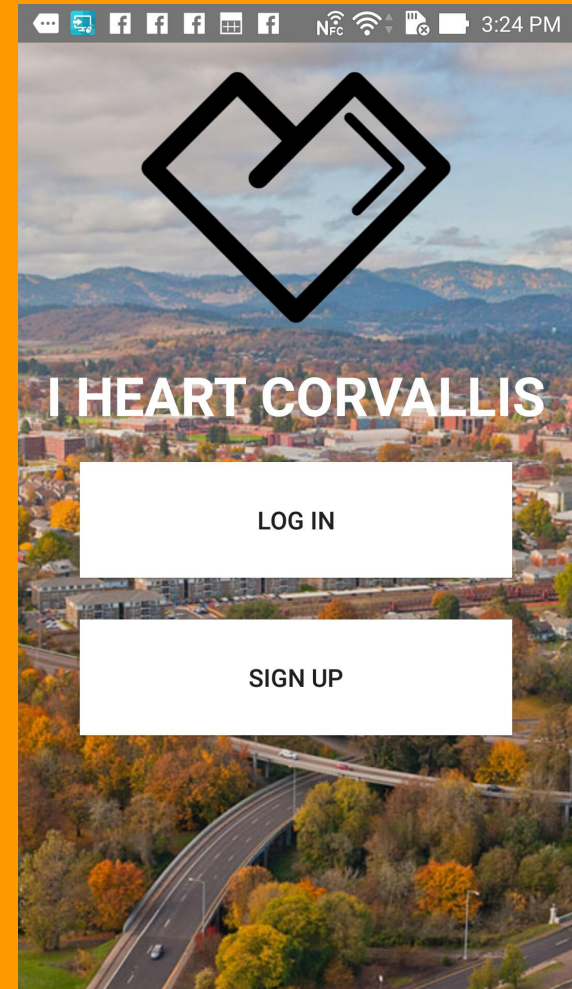
- The app will act as a passport for students
- Passport will initially be empty but will become populated with stamps as the user attends specific events around Corvallis
- Some events will be worth more stamps than others
- When a user accumulates enough stamps, they will get a notification that they are eligible for an award
- Real rewards that will be given out at the Office of Student Life

DESIGN BREAKDOWN RECAP

- Responsibilities split among group
- Omeed: Data Storage and Handling, UI Design Architecture, and Administrative Access
- Dylan: User Security, Interface Scaling, and Application Testing
- Bradley: Splash screen, Geolocation Services, and Event Display, Admin website (client side)

WELCOME PAGE

- Animated splash screen opens upon app startup
- Users can log in or sign up
 - Logo will change to the IHC logo
- If the user is already logged in, the app will skip straight to the dashboard



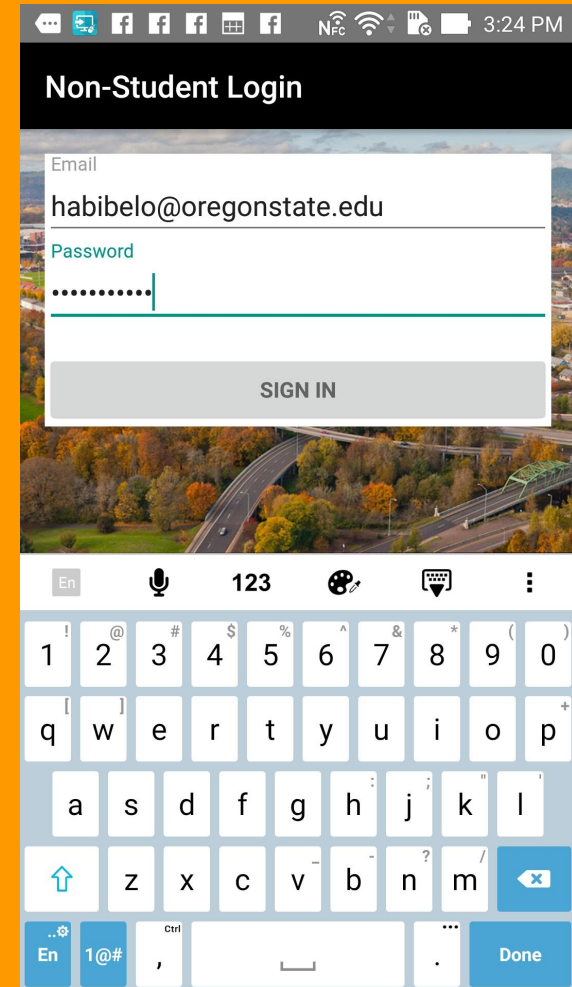
SIGNUP PAGE

- User enters first and last name, email, and password
- Page ensures email and password are in the right format
- Gives user an error alert if something has been input incorrectly, or if that account already exists
- Adds account to database if that account does not already exist and values are in the correct format
- OSU students can tap the grey text box to skip this signup and log in with their ONID account

The screenshot shows a mobile application interface for signing up. At the top, a black header bar contains the text "Sign Up" in white. Below this, a grey rectangular box contains the text "ARE YOU AN OSU STUDENT? LOG IN WITH YOUR ONID BY TAPPING HERE!". Underneath the grey box, there are four input fields: "First Name" (containing "Omeed"), "Last Name" (containing "Habibelahian"), "Email" (empty), and "Password" (empty). The background of the app is a scenic view of a campus. At the bottom, a virtual keyboard is visible, with a "Next" button in the bottom right corner.

LOGIN PAGE

- Adds user name/email to session
 - Soon to include other info
- Checks database for matching email/password combination



The image shows a mobile application interface for a "Non-Student Login". The screen features a dark header with the title "Non-Student Login". Below the header, there is a form with two input fields: "Email" and "Password". The "Email" field contains the text "habibelo@oregonstate.edu". The "Password" field is masked with dots. A "SIGN IN" button is located below the password field. The background of the app is a scenic image of a road winding through a forest. At the bottom of the screen, there is a virtual keyboard with a "Done" button.

Non-Student Login

Email
habibelo@oregonstate.edu

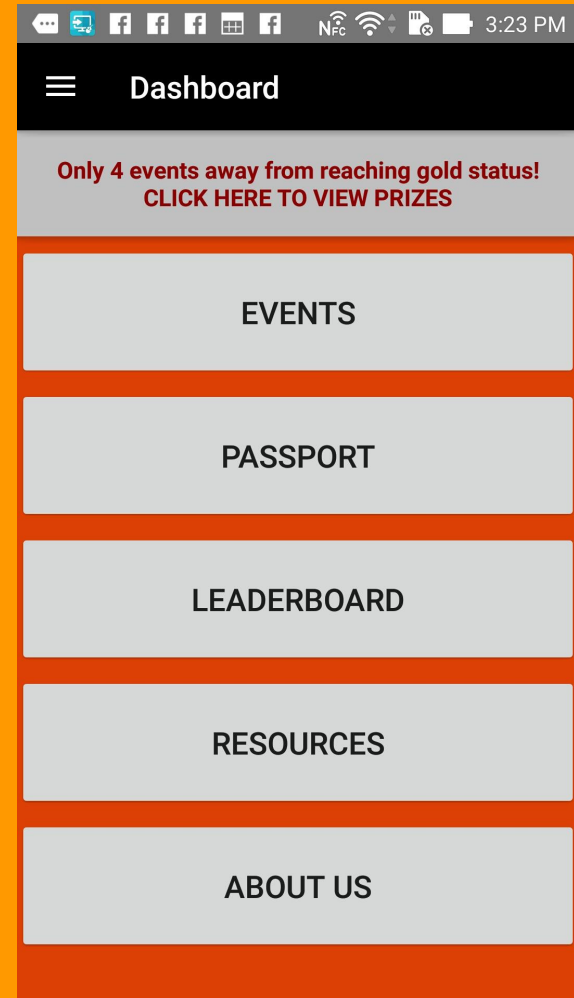
Password
.....

SIGN IN

En 123 1@# , _ . Done

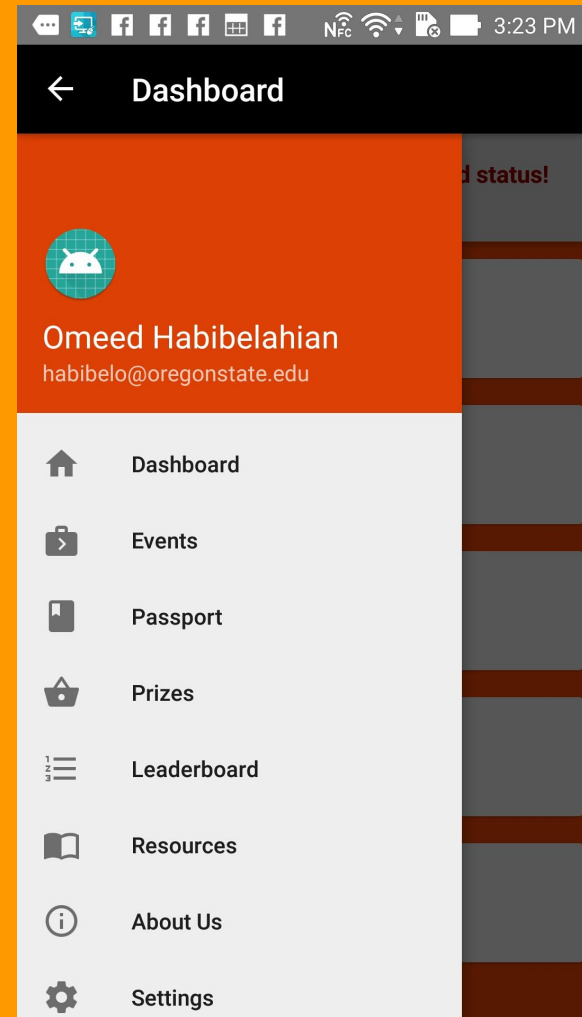
DASHBOARD

- Currently displays buttons that link to other pages
- Will eventually pull info relevant to user from other pages
 - Intended to act like a news feed for the user



NAVIGATION MENU

- Click to go to any page
- Displays username/email
 - Uses session stored info to display



EVENTS PAGE: LIST VIEW

- Displays upcoming events around Corvallis
- Click on each tile to get more information on an event



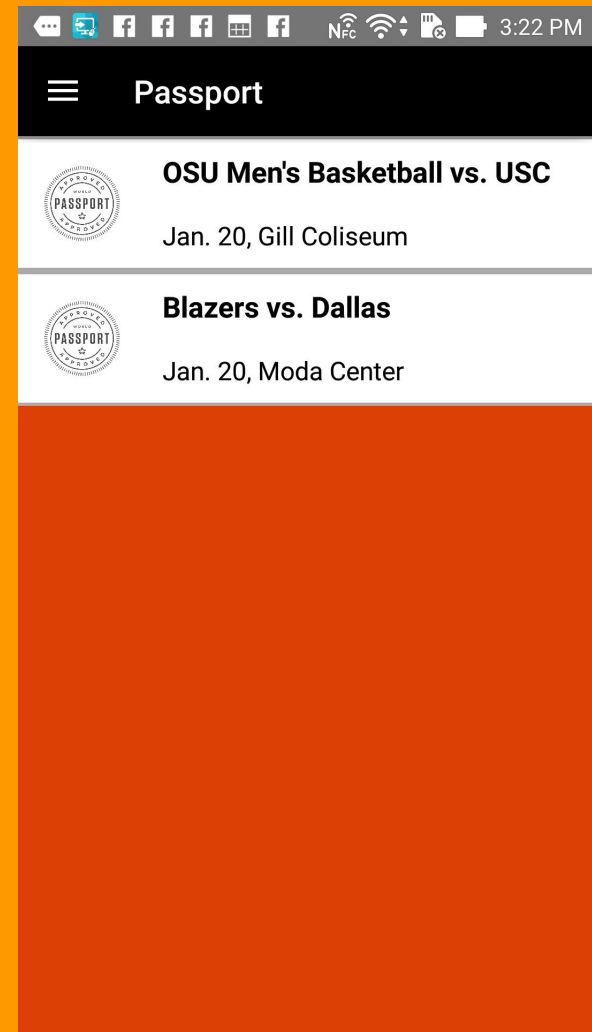
EVENTS PAGE: CARD VIEW

- Click button to change viewtype
- Includes a picture of the event provided by the event host
- Need to implement “learn more” page



PASSPORT

- Displays the user's completed events
 - We will eventually design our own stamps
- Displays events by the date completed



LEADERBOARD

- Lists each registered user sorted by the number of stamps they've received
 - We will eventually cap the number of users shown

A screenshot of a mobile application's leaderboard screen. The screen has a dark grey header with a hamburger menu icon and the title "Leaderboard". Below the header is a table with two columns: "NAME" and "STAMPS". The table lists five users: Lyndi Petty (16 stamps), Dylan Tomlinson (15 stamps), Omeed Habibelahian (13 stamps), Bradley Imai (13 stamps), and brad imai (0 stamps). Below the table is a solid orange-red rectangular area.

NAME	STAMPS
Lyndi Petty	16
Dylan Tomlinson	15
Omeed Habibelahian	13
Bradley Imai	13
brad imai	0
ADAM Bartlett	0

PRIZE PAGE

- Lists prizes users can get
- Will display eligible prizes
- Prizes broken down into gold, silver, and bronze levels, each level requiring more stamps to reach
- User can win better prizes by completing more events and accumulating more stamps



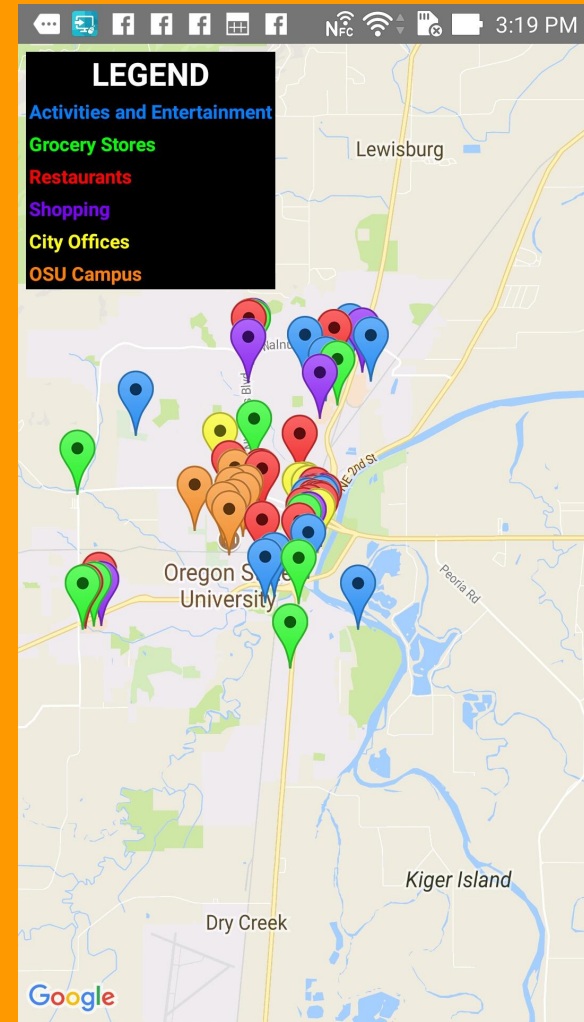
RESOURCES

- Will hold links to useful resources around Corvallis



RESOURCE MAP

- Shows notable locations of Corvallis
- Clicking on a location will give more details about it



CLIENT WEBSITE

I HEART CORVALLIS

Administrative Suite

[Home](#)[Events](#)[Resources](#)[Prizes](#)[Add an Event](#)[Manage Events](#)[Manage Primary Resources](#)[Manage Resource Map](#)[Add a Prize](#)[Manage Prizes](#)

CLIENT WEBSITE: ADD AN EVENT

Add an Event

* Denotes a required field

* Name of Event:

* Name of Location:

* Street Address:

* City:

* State:

* ZIP:

* Date:

* Time:

* Description of Event:

* Cover Image: No file chosen

Link 1:

Link 2:

Link 3:

Event PIN:

CLIENT WEBSITE: MANAGE EVENTS

Manage Events

Name	Location	Date and Time	Action
OSU Men's Basketball vs. USC	Gill Coliseum	2018-01-20 17:00:00	Edit Delete
Blazers vs. Dallas	Moda Center	2018-01-20 19:00:00	Edit Delete
Blazers vs. Minnesota	Moda Center	2018-01-24 19:00:00	Edit Delete
OSU Men's Basketball @ Oregon	Matthew Knight Arena	2018-01-27 17:00:00	Edit Delete
60th Grammy Awards	Madison Square Garden	2018-01-28 16:30:00	Edit Delete
Blazers vs. Chicago	Moda Center	2018-01-31 19:00:00	Edit Delete
OSU Men's Basketball vs. WSU	Gill Coliseum	2018-02-08 19:30:00	Edit Delete

COMMUNICATING WITH THE DATABASE

```
class NonStudentAuthProcess extends AsyncTask {

    private Context context;
    private String email;
    private String password;
    final static String IHC_NS_LOGIN_URL = "http://web.engr.oregonstate.edu/~habibelo/ihc_server/appscripts/non_student_login.php";

    public NonStudentAuthProcess(Context context) { this.context = context; }

    protected void onPreExecute() { showProgress(true); }

    @Override
    protected Object doInBackground(Object[] objects) {
        email = (String) objects[0];
        password = (String) objects[1];

        try {
            URL url = new URL(IHC_NS_LOGIN_URL);
            String data = URLEncoder.encode( s: "email", enc: "UTF-8" ) + "=" + URLEncoder.encode(email, enc: "UTF-8");
            data += "&" + URLEncoder.encode( s: "password", enc: "UTF-8" ) + "=" + URLEncoder.encode(password, enc: "UTF-8");

            HttpURLConnection conn = (HttpURLConnection) url.openConnection();

            conn.setRequestMethod("POST");
            conn.setDoOutput(true);
            OutputStreamWriter wr = new OutputStreamWriter(conn.getOutputStream());
            wr.write( data );
            wr.flush();

            BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));

            StringBuffer sb = new StringBuffer("");
            String line = null;

            while ((line = reader.readLine()) != null) {
                sb.append(line);
                break;
            }

            return sb.toString();
        } catch (Exception e) { return new String( original: "Exception: " + e.getMessage()); }
    }












    @Override
    protected void onPostExecute(Object result) {
        String resultString = (String) result;
        NonStudentLoginActivity.this.onBackgroundTaskDataObtained(resultString);
    }
}
```

COMMUNICATING WITH THE DATABASE

```
private void onBackgroundTaskDataObtained(String result) {
    if (result.contains("LOGINSUCCESS")) {
        try {
            Thread.sleep( millis: 3000);
            //tokenize through result string to get name, email, etc.
            //session.createLoginSession("Omeed Habibilian","")
            StringTokenizer st = new StringTokenizer(result);

            String first = st.nextToken( delim: "\\");
            first = st.nextToken( delim: "\\");
            String last = st.nextToken( delim: "\\");
            String name = first + " " + last;
            String email = st.nextToken( delim: "\\");
            session.createLoginSession( name , email);
            Intent intent = new Intent( packageContext: NonStudentLoginActivity.this, DashboardActivity.class);
            startActivity(intent);
        } catch (Exception e) {
            Toast.makeText( context: this, text: "Exception" + e.getMessage(), Toast.LENGTH_LONG).show();
        }
    }
    else {
        mEmailView.setText("");
        mPasswordView.setText("");
        showProgress(false);
        AlertDialog.Builder builder;
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            builder = new AlertDialog.Builder( context: this, android.R.style.Theme_Material_Dialog_Alert);
        }
        else {
            builder = new AlertDialog.Builder( context: this);
        }
        builder.setTitle("Login Error");
        builder.setPositiveButton( text: "Try Again", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                // Try logging in again
            }
        });
        if (result.equals("AUTHERROR")) {
            builder.setMessage("Incorrect email/password combination.");
        }
        else if (result.equals("DUPACCOUNTERROR")) {
            builder.setMessage("Unable to authenticate. More than one account exists with this email/password combination.");
        }
        else if (result.equals("SEARCHERROR")) {
            builder.setMessage("Error searching for account.");
        }
        builder.setIcon(android.R.drawable.ic_dialog_alert);
        builder.show();
    }
}
```

DATABASE SCHEMA: USERS

←T→			student	firstname	lastname	email	password	id	stampcount	profilepic	age	race
<input type="checkbox"/>			0	Omeed	Habibelahian	habibelo@oregonstate.edu	HIDDEN	1	13	[BLOB - 0 B]	22	NULL
<input type="checkbox"/>			0	Bradley	Imai	imaib@oregonstate.edu		2	13	[BLOB - 0 B]	NULL	NULL
<input type="checkbox"/>			0	Dylan	Tomlinson	tomlinsd@oregonstate.edu		3	15	[BLOB - 0 B]	NULL	NULL
<input type="checkbox"/>			0	Lyndi	Petty	lyndipetty@gmail.com		4	16	[BLOB - 0 B]	NULL	NULL
<input type="checkbox"/>			0	brad	imai	jasdfja@gmail.com		5	0	[BLOB - 0 B]	NULL	NULL
<input type="checkbox"/>			0	ADAM	Bartlett	bartlead@oregonstate.edu		6	0	[BLOB - 0 B]	NULL	NULL

RETRIEVING DATA FROM THE DATABASE: LOGIN

```
17 if ($_SERVER["REQUEST_METHOD"] == "POST") {
18     $email = $_POST['email'];
19     $password = $_POST['password'];
20
21     $result = $mysqli->query("SELECT email, password FROM ihc_users");
22     if ($result->num_rows > 0) {
23         while ($user = $result->fetch_assoc()) {
24             if ($user["email"] == $email && $user["password"] == $password) {
25                 $isAuth = True;
26                 break;
27             }
28         }
29     }
30     if ($isAuth == True) {
31         $result = $mysqli->query("SELECT firstname, lastname, email, id FROM ihc_users WHERE email='$email' AND password='$password'");
32
33         if ($result->num_rows == 1) {
34             while ($row = $result->fetch_assoc()) {
35                 echo "LOGINSUCCESS\\\\";
36                 echo $row["firstname"] . "\\\\";
37                 echo $row["lastname"] . "\\\\";
38                 echo $row["email"] . "\\\\";
39                 echo $row["id"] . "\\\\";
40                 echo ";;";
41             }
42         }
43         else if ($result->num_rows == 0) {
44             echo "NOACCOUNTERROR"; # account does not exist
45         }
46         else if ($result->num_rows > 1) {
47             echo "DUPACCOUNTERROR"; # more than one account with this email/password
48         }
49         else {
50             echo "SEARCHERROR"; # error looking for account
51         }
52     }
53     else {
54         echo "AUTHERROR"; # unable to authenticate email/password
55     }
56 }
```

SESSION MANAGEMENT

```
public void logoutUser(){

    //clearing all data from shared preferences
    editor.clear();
    editor.commit();

    //after logout redirect user to main activity
    Intent i = new Intent(_context, MainActivity.class);

    //closing all the activities
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

    //add new flag to start new activity
    i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

    //starting login activity
    _context.startActivity(i);
}

//Quick check for login
public boolean isLoggedIn() {

    return pref.getBoolean(IS_LOGIN, false);
}
```

```
public void checkLogin(){

    if(!this.isLoggedIn()){
        // user not logged in redirect to dashboard
        Intent i = new Intent(_context, MainActivity.class);

        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

        i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        _context.startActivity(i);
    }
    else {
        Intent i = new Intent(_context, DashboardActivity.class);

        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

        i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        _context.startActivity(i);
    }
}
```


SESSION CREATION

```
private void onBackgroundTaskDataObtained(String result) {  
    if (result.contains("LOGINSUCCESS")) {  
        try {  
            Thread.sleep(3000);  
            //tokenize through result string to get name, email, etc.  
            //session.createLoginSession("Omeed Habibilian","")  
            StringTokenizer st = new StringTokenizer(result);  
  
            String first = st.nextToken("\\\\");  
            first = st.nextToken("\\\\");  
            String last = st.nextToken("\\\\");  
            String name = first + " " + last;  
            String email = st.nextToken("\\\\");  
            session.createLoginSession( name , email);  
            Intent intent = new Intent(NonStudentLoginActivity.this, DashboardActivity.class);  
            startActivity(intent);  
        } catch (Exception e) {  
            Toast.makeText(this, "Exception" + e.getMessage(), Toast.LENGTH_LONG).show();  
        }  
    }  
    else {
```

PROBLEMS WE HAVE ENCOUNTERED

- Finding a long-term host for the database
- Communicating data between the server and the device
- Client continues to bring up small changes throughout the application
- Unknown if we have permission from the university to utilize CAS (ONID) login system

SOLUTIONS TO PROBLEMS

- Met with Derek Whiteside, the Director of Web and Mobile Services for OSU
 - Discussed hosting our database on their servers
 - Gave lots of good feedback in regards to application
- Researched how Android applications communicate with databases, applied that knowledge to our application
- Weekly meetings with Client to discuss current and future design changes
- Planning to meet up with Kegan Sims to discuss OSU branding
 - If we cannot use CAS (ONID) login, will look into facebook login or other options

TO DO: USER SECURITY

- Implement the hash function Password-Based Key Derivation Function 2 (PBKDF2) for User Passwords
- Sanitize user inputs
 - Protect from SQL injections
- Implement techniques to protect PHP scripts

TO DO: INTERFACE SCALING

- App will need to run on a variety of different Android devices
 - All have different screen sizes, resolutions, and dpi
- Important to ensure our application retains its visual quality regardless of which device it is running on

TO DO: GEOLOCATION SERVICES

- Retrieving current geolocation
- Integrating Google Maps API
 - Allow user to view events
 - Authentication that user was there
- Connection to other parts
 - Will need to access the event database

TO DO: DATA STORAGE & HANDLING

- Encode our PHP outputs in JSON
 - Currently passing strings with User info
 - Allows for more flexible manipulation of data
- Updating the fields in each table
 - We do not know the exact data we will need for each event, Lyndi will soon inform us of what is required
 - Fields will change as our application reaches production stage and our needs change

TO DO: UI DESIGN

- Continuous UI changes
 - As features are added, the user interface has to change
 - Meeting weekly with client as these changes are made
- Need interface to be intuitive, easy to use

TO DO: ADMINISTRATIVE ACCESS

- This is basically how the CCR will control in-app content
- CCR office is not very technical and need to be able to have some sort of control of content within the application
- They will need to be able to
 - Add, modify, and remove events from the event database
 - Give or take stamps from users
- We will build the CCR an administrative account and interface that gives them this functionality

PROGRESS OVER WINTER BREAK

Deltas

- Begin implementation of the application

Actions

- First began implementation with HTML and PHP
- Later, figured that this route made connection to the database too difficult to implement and began to use Java and XML in order to create the Android layout

Positives

- Completed a skeleton application to showcase to our clients
- Created login page, dashboard page and pages that are linked from the dashboard page

REPORTS FROM WEEKS 1-2

Deltas

- Set a consistent meeting time with our client for next term

Actions

- Once Winter term starts, talk to our clients about the best time to meet with them to discuss app progress
- Setting up weekly blocks of time to work on the app

Positives

- Contacted client and set up a meeting date
- Showcased Mobile application we created over winter break
- Splash Screen
- More UI changes for prototype
- Created Google map with notable landmarks of Corvallis
-

REPORTS FROM WEEKS 3-5

Deltas

- Continuously make progress on application

Actions

- Meeting several times a week to work on the app

Positives

- RecyclerView Layout + Card View
- Connected database to application
- Admin website to modify data entries
- More UI changes
- Session Management