Introduction:

The Bsu_Campus_App is a free Android application for mobile devices, designed to be a useful tool that can be used by the Bridgewater State University community to access important information about places and events going on the BSU Campus.

In the recent years the mobile device field has been growing exponentially around the world; devices such as Smartphones and tablets have become part of our daily life. Web browsing, shopping, money orders and transfers, job applications, videoconferences, etc, are some of the numerous activities that are currently performed using mobile devices.
Google Android [1] is one of the most competitive among the current mobile device Operating Systems; the archrival of Apple’s IOS [2] in the USA and around the world. Google announced last February during the Mobile World Congress that its mobile apps marketplace has more than 450,000 apps now [3].

Figures 1 and 2 show the Android daily activation rate and an estimate of the total of Android activations by the end of 2013. Last November, Google was already activating Android devices in a rate close to 900,000 a day. By the end of 2013 Google expects to have a total of 1 billion of Android device activations. [4]

Fig. 1: Android Daily Activation Rate
Following the worldwide growth in the mobile device field, the number of smartphone users within the BSU community has also been increasing. This project is intended to provide an app that allows faculty, students and staff to have useful information about the BSU campus in their Smartphones. In the USA, nearly half of all smartphones are owned by the 18 to 34 demographic. Approximately 18 million college students own smartphones across the country and this fact has accelerated the development of apps designed specifically for students. [8]
Similar Work

As the mobile device field grows, a lot of applications in many different areas have been developed. [1][2] This has extended the limits of the mobile app field making the process of getting information on the go much easier. In the academic area, some notable applications are already available to customers. Some examples of these applications are listed below:

- **MIT Mobile App:**

  This is a free mobile application available for both Android and IOS users. This application has been collaboratively designed and developed by MIT students and staff in a group headed by Marilyn T. Smith, head of Information Services and Technology, to extend the Institute’s mobile framework.

  This app is similar to the Bsu Campus App since it provides users with the latest news about research and innovation, class information and announcements and emergency information in this case about the MIT.

  In addition to the common facilities between these two apps, the MIT Mobile App offers some extra features. These features include real-time shuttle tracking and alerts, an interactive campus map, a searchable staff directory, a self-guided campus tour, emergency information, MIT Libraries account management and search, and a way to report campus repair and maintenance issues. [5]

  The BSU Campus App, on the other hand, is focused on the BSU community and provides users with information about the interior of the BSU Campus buildings from any place and also automatically displays related information when users are in the close vicinity of these buildings.
➢ **TAMUmobile Apps:**

Developed by Texas A&M, TAMUmobile App is a free mobile application developed with the goal of providing users with access to information on and about Texas A&M. This App provides information about a number of campus offices, departments and divisions and services at the Texas A&M campus.

This App shares some common features with the BSU Campus App such as campus news, directories and events. In addition to these features TAMUmobile App provides access to campus maps and course information and is available for Android, iPhone/iPodTouch and Blackberry. [6]

In this case, the BSU Campus App also has the advantage of providing users with information about the interior of the buildings. This feature helps users know exactly what they can find inside each of the campus buildings. This information is also automatically provided when users are in the vicinity of any of the BSU Campus Buildings.

➢ **College Confidential Mobile App (CC App):**

Developed by the College Confidential Team, this mobile app for iPhone and Android shares some similarities with the BSU Campus App, since both of them provide continued mobile information about Colleges. However, the CC App is more focused on implementing access to services that are common to any College instead of focusing on a specific School as the BSU Campus App does.
Examples of these services are the CC forums, Ask the Dean, School Spotlight blogs and CampusVibe. [7]

- **Georgia College Mobile App:**

  By the end of the Spring 2012 semester, Georgia College expects to announce their GC Mobile App. This mobile App will incorporate Georgia College specific services including myCats, GeorgiaVIEW, a campus directory and an Athletics portal that will provide news and score updates during sporting events. [8]

  At Georgia College, myCATS is the main virtual location to reach college-related information that resides on the web. It allows users to check e-mail, register for courses, communicate with groups, post to users’ electronic calendar, etc. GeorgiaVIEW is a Learning Management System used in the University System of Georgia. It is a helpful tool for web-based courses that provides online access to features such as syllabus, linked pages, e-mail, online testing, gradebooks, chat rooms blogs and journals. [9]

  The incorporation of myCats and GeorgiaVIEW will help students to manage class information, contact classmates and professors, review degree and graduation information and even add and drop classes all from their mobile device.

  This application is also expected to provide quick response codes on fliers and posters on campus to give students additional information on their electronic
devices. By scanning these codes, students will get contacts and locations information about different departments and professors on their phones.

The GC mobile App will be similar to the BSU Campus App since it is expected to provide information and access to school features. However, the GC mobile does not exist yet while the BSU Campus App is already available.

The following apps are three of the most widely shared free mobile applications between college students: [10]

- **Evernote:**

  This app popular among college students is a useful tool not only for its note-taking functions but also for its cross-platform capabilities, allowing users to clip pictures, sound files, videos, and pages from multiple sources and access them from their phone.

- **Wi-Fi Finder:**

  This is a mobile app developed for Android and IOS able to scan for Wi-Fi hotspots around users and automatically connect to them. It allows users to search results as a list or on a map without leaving the app. With this app it is also possible to Get Wi-Fi hotspot details, call the hotspots locations, get directions and add them to users’ favorites.
➢ **Urbanspoon:**

This app is known as the hungry college student’s favorite app. For users who love finding new places to eat, Urbanspoon provides a random selection feature. For those who are on a budget, have specific preferences, or don’t want to walk too far, Urbanspoon allows customizations of search results by price, type, and distance. The results are displayed with a shake of the phone.

These last three apps are related to the BSU Campus App since they are also used by members of school communities. However, while the BSU Camps App is focused on the Bridgewater State University these three apps are more general and intended to be used by members of any school community.

---

**BSU Campus App Operation**

When users click the BSU CAMPUS icon in their device, the application starts and the Bsu_Campus_App User Interface (main interface) will be displayed. At this point, the app will keep waiting for the occurrence of one of the two events that can change its status. These events are:

➢ OnLocation Changed

➢ OnClick
OnLocation Changed

When the location of the device changes, the application will retrieve the new location and compare it with the location of each one of the BSU Campus buildings. If the new location coincides with the location of any of these buildings, the User Interface of this corresponding building will be displayed replacing the previous User Interface. For example, if the device is in the vicinity of Boyden Hall, the Boyden User Interface is shown.

The Boyden User Interface is similar to the other buildings’ User Interfaces. These UIs provide users with an initial information page with a picture of the corresponding building and some quick information related with this building. The buildings’ UIs also have three buttons at the bottom part that allows users to interact with the application. These buttons are:

- St. Ann
- More
- Home

When the St. Ann Button is pressed, the Students Announcements activity starts, and the current student announcements from the BSU web site will be displayed on the device.

When the More button of a specific building UI is pressed, the more User Interface of that specific building is shown and information about the interior of that building will be displayed.
When pressed, the Home button restarts the Bsu_Campus_App activity, thus the App main interface will be displayed again.

**OnClick**

While at the main interface, users are provided with the following buttons that can be pressed in order to interact with this app.

- BSU Colleges
- Student and Employee Email
- Info Bear
- Student Announcements
- BSU Halls
- Events Calendar
- Class Cancelation

When users press the BSU Colleges, Student and Employee Email and Info Bear buttons the respective BSU web page are loaded in Users devices.

On the other hand, when the Student Announcements button is pressed the Student Announcements Activity starts and displays the current student announcements from the BSU web site at that moment.

The same process occurs when the Events Calendar and Class Cancelation buttons are pressed. In these cases, the current events at the BSU Site Calendar and the class cancelations respectively will be displayed.
If the Halls button is pressed, the Halls User Interface generated by the Halls.xml file will be shown, providing users with the possibility of clicking one of the available building icons. The Halls UI provides users with information about important places on the BSU Campus. By pressing a building icon, that building UI will be shown, providing users with information about that specific building.

Figure 3 describes the entire structure of this application.

Fig. 3: The Bsu_Campus_App structure
Structure of the Bsu_Campus_App

An Activity is a special type of Android Class that represents the tasks that users can do, thus Activities are focused on interacting with users. Activities are initialized using the onCreate() method while the setContentView() method is used to connect activity classes with user interfaces. The findViewById() is also used to retrieve the components of the UI that users need to interact with programmatically.

An Activity in Android has the following main states [14]:

- Starting
- Running
- Paused
- Stopped
- Destroyed or Killed

Starting state

When an activity that doesn’t exist in memory is called, it goes through the starting state until it reaches the running state.

This transition from the starting state to the running state is one of the most expensive operations in terms of computing time, and this also directly affects the battery life of the device. Because of these, Activities that are no longer shown are not automatically destroyed thus users can quickly go back to them without having to go through the entire starting stage.
Running state

The activity in the running state or in focus is the one that is currently on the screen and interacting with the user. All user interactions such as typing, touching the screen, and clicking buttons are handled by this one activity.

There is only one running activity at any given time. The running activity is the one that has priority in terms of getting the memory and resources it needs to run as quickly as possible.

Paused state

When an activity is not interacting with the user but still visible on the screen, it is in the paused state. This stage is not a typical scenario since mobile device’s screens are usually small and an activity is either taking up the whole screen or none at all.

However, the paused state is often seen in the cases of dialog boxes that come up in front of an activity causing it to become paused. Paused activities still have high priority in terms of getting memory and other resources.
**Stopped state**

When an activity is not visible, but still in memory, it is in the stopped state. A stopped activity can go back to the running state again or become destroyed and removed from memory.

The system keeps activities around in a stopped state because it is likely that the user will still want to get back to those activities sometime soon, and restarting a stopped activity requires fewer resources than starting an activity. That is because all the objects are already loaded in memory and simply have to be brought to the foreground. Stopped activities can be removed from memory at any point.

**Destroyed or killed state**

A destroyed activity is no longer in memory. The Activity Manager can decide that a stopped activity is no longer needed and remove it from memory. It is possible for a paused activity to be destroyed as well.

Activities in the system are managed in an activity stack. When an activity is running, it is placed on the top of the stack and any previous activity will now be below it. The application on the top of the stack has the highest priority from the operating system.
The Bsu_Campus_App is composed by the three different types of components shown below:

- Activities (Bsu_Campus_App)
- User Interfaces (mainboyden)
- Resources (boydenicon.png)

Figure 4 shows the relationships among these three types of components in this application.

![BSU Campus App Activity](image)

Fig. 4: BSU Camps App Components

All the Activities belonging to the whole project are implemented in .java files while the UIs are implemented in .xml files.

There are also four different types of Activities in this application with the following designations:

- Bsu_Campus_App
- Halls
The Bsu_Campus_App Activity:

The Bsu_Campus_App Activity is the main Activity of the whole application. Figure 5 shows the User Interface for this Activity that is specified by the main.xml file.

![Fig. 5: Bsu_Campus_App User Interface](image)

This activity is divided into two parts. The first half is dedicated to the event handlers for starting the StudentsAnnouncements, EventsCalendar, ClassCancelations and Halls Activities and showing the BSU Colleges’ web pages, the InfoBear and the Student and Employee email when the appropriate buttons are pressed. Figures 6 and 7 show the source code used to start the StudentAnnouncements Activity and show the web site of
the BSU College of Graduate Studies respectively when the appropriated button is clicked.

Fig. 6: Starting the StudentAnnouncements Activity

```java
Button tostudent = (Button) findViewById(R.id.mainbutton2);
tostudent.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        Intent myIntent = new Intent(view.getContext(), StudentAnnounce.class);
        startActivityForResult(myIntent, 0);
    }
});
```

Fig. 7: Displaying the BSU College of Graduate Studies web page

```java
Button tograduate = (Button) findViewById(R.id.mainbutton9);
tograduate.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        String url = "http://www.bridgew.edu/CoGS/";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }
});
```

The second half of the Bsu_Campus_App activity implements the java LocationListener Class. The onLocationChanged method of the LocationListener Class retrieves devices' Latitude and Longitude values. These values of Latitude and Longitude are then used to determine whether or not a device is in the vicinity of a BSU Campus building and then
display the correct UI. This UI provide users with information related to their specific location on campus.

Due to the proximity between some of the BSU Campus buildings, some of them are not used in this part of the application. Thus, the Hart and Burnel halls and the Kelly Gym are not used due to their close location to the Moakley Center and the Maxwell Library respectively.

Figure 8 shows part of the code that is executed when users are in the vicinity of the Moakley Center.

```java
if (loc.getLatitude() > 41.98822) && loc.getLatitude() < 41.98913) {
    if (loc.getLongitude() > -70.96692) && loc.getLongitude() < -70.95583) {
        if (moakleyFlag == 1) {
            moakleyFlag = 0;
            setContentView(R.layout.mainmoakley);

            Button tomackleyevents = (Button) findViewById(R.id.mainmoakleybutton1);
            tomackleyevents.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent myIntent = new Intent(view.getContext(), StudentAnnounce.class);
                    startActivityForResult(myIntent, 0);
                }
            });

            Button toremoakley = (Button) findViewById(R.id.mainmoakleybutton2);
            toremoakley.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent myIntent = new Intent(view.getContext(), MoreMoakley.class);
                    startActivityForResult(myIntent, 0);
                }
            });
        }
    }
}
```

Fig. 8: Moakley location handler
**Parsing Activities**

Included in the Parsing Activities category are the three Activities showed below:

- StudentAnnouncements
- ClassCancel
- EventsCalendar

The StudentAnnouncements Activity uses the Simple API for XML (SAX) [11] in order to parse information from the Student Announcements [12] page of the BSU Site. This information to be parsed is stored in the BSU Site using the Really Simple Syndication document format (RSS Feed) [13].

The openStream() method is used to input the XML readings. These XML readings include RSS Feeds title, publication date, description and link.

The ClassCancel and EventsCalendar Activities follow the same structure of the StudentAnnouncements Activity. The only difference among these Activities is the RSS Feed sources. The ClassCancel and EventsCalendar Activities read RSS Feeds from the Class Cancelations and Events Calendar pages of the BSU Site respectively.

The Layouts for these three last Activities are dynamically created by the Activities while the device is reading the content from the BSU Site.

The implementation of layouts in the same file as the Activities is discouraged by the android developers’ community. However, in such cases where the layout changes dependent on the content of other applications, which are not part of the project itself,
this approach is needed because Android .xml based UIs are static and can’t be changed during execution time.

**Building’s Information Activities:**

The Buildings’ Information category is formed by Activities such as the MoreMoackley, the MoreBoyden, the MoreRCC, etc. These are simple activities that provide users with more information about the configuration of each building. This information mainly lists the public spaces in a building, including the offices, auditoriums, classrooms, and other specific places. They are static Activities that are started when the More button at the interface of each building is pressed. The UIs for these Activities are generated separated in .xml files using the common features of the android layouts such as textviews, imageviews, scrollviews and linear layouts.

**Project Layout**

The most common and most recommended way of creating a user Interface for Android applications is to write the layout codes in separate .xml files. Following this approach, both the activity and its layout are implemented in two different kinds of files and then connected during the execution time. In order to connect a .xml layout file to any Activity, the setContentView() method must be called and a reference to the .xml root node must be provided.
In this project, the primary layout of the Bsu_Campus_App activity shown in figure 5, which is the main activity of the whole project, is implemented in the main.xml file. However, as mentioned before, when the device is in the vicinity of any of the BSU Campus buildings, the interface is automatically changed to the one corresponding to that building.

A View in Android is a class that represents the basic building block for User Interface components. It occupies a rectangular area on the screen and is responsible for drawing and event handling. ScrollViews, TextViews and ImageViews are subclasses of the View class that are used in this application to scroll the UI down and up, display texts and display images on devices screen.

Multiple Views can be combined forming ViewGroup and LinearLayouts. A LinearLayout is a ViewGroup that displays View elements in a linear direction, either vertically or horizontally.

In Android a Style is a collection of properties that specify the look and format for a View. A style must be defined in a .xml file separated from the layout file and can specify properties known in Android as attributes such as height, padding, font color, font size, background color and text gravity. On the other hand, a Theme in Android is a style that is applied to an entire activity or project rather than a single view. When a Style is applied as a Theme to an Activity, every View in that Activity will apply that Style. The selection of Themes for activities is done via the project manifest.xml file.

Themes, Styles and Attributes are used in the layout of this application to improve the look of the LinearLayouts, ScroolViews, TextViews, ImageViews and Buttons.
Project User Interfaces

This application is formed by a set of User Interfaces grouped into the four different categories described below:

- Main UI
- Buildings UIs
- Buildings’ Information UIs
- Parsing UIs

Main UI

The Main UI is the main User Interface of the BSU_Campus_App. In this UI, the device’s display is implemented with a vertical LinearLayout which is then divided into three vertical ScrollViews. These ScrollViews are also constituted by vertical and horizontal LinearLayouts where all the ImageViews, TextViews and Buttons are placed.

The Main UI uses the styles belonging to the Theme applied by the Bsu_Campus_App Activity.
Figure 9 shows the main UI of the project and part of the main.xml file used to generate this UI.

![Main UI and main.xml file](image)

**Fig. 9:** The Project main UI and the main.xml file

**Buildings UIs**

Buildings UIs such as mainboyden, mainmoackley and mainecc are the User Interfaces that are shown to users when they are in the proximity of the respective building. These UIs also use the styles applied in the Theme used by the Bsu_Campus_App activity. These UIs follow a structure similar to the Main UI since they all implement the same kind of features.

Figure 10 shows the Boyden Hall UI and part of the mainboyden.xml file used to generate this User Interface. In the whole application, each building has its own .xml file that generates its UI.
Buildings UIs provide users with quick information related to each specific building, while still allowing them to read the current Student Announcements or to search more information about a specific building by pressing the buttons located at the bottom part of the User Interface.

**Buildings’ Information UIs**

Buildings’ Information UIs such as morehart, morercc and moreboyden that is shown in figure 11 are used by the Buildings’ Information Activities, in order to display information about the interior of the BSU Campus buildings. For example, when the MoreBoyden Activity is started, it automatically connects to the moreboyden UI and provides users with information about the location of offices such as the Students Account Office, the
Registrar Office, the Systems office, etc. These UIs have a simple structure with no Theme at all, only using a LinearLayout with TextViews to show the correct information.

Figure 7 shows the Moreboyden UI and part of the .xml file used to generate it.

```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="BOYDREN HALL"
        android:layout_gravity="center"
        android:textAlign="center"
        android:textStyle="bold"
        android:textSize="25sp"
        android:textColor="#666666"
        android:shadowColor="#666666"
        android:shadowRadius="3.5"
        android:shadowDx="3"
        android:shadowDy="3"
    />
    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1">
```

Fig. 11: Moreboyden UI and the moreboydren.xml file used to generate it.

**Parsing UIs**

The Parsing UIs category includes the User Interfaces used to display the information parsed from the BSU Site by the StudentAnnouncements, ClassCancelations and EventsCalendar Activities. These UIs are created dynamically by these Activities and vary in dependence of the content of the pages from where the information is parsed.
**Themes**

In Android, all Themes used in an application have to be implemented in a separate .xml file and saved in the project Values folder. In this project, a single theme called Theme is implemented in the themes.xml file. This Theme is then used by the Bsu_Campus_App.java Activity. Figure 8 shows part of the themes.xml file where the theme called Theme is implemented.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="Theme" parent="android:Theme">
    <item name="android:windowTitleSize">20dp</item>
    <item name="pageMargin">2sp</item>
    <item name="whiteBackground">@style/white_background_wh</item>
  </style>
</resources>
```

Fig. 8: The themes.xml file

**Styles.xml**

Android also requires the implementation of Styles in a separate .xml file. In this project all Styles are stored in the styles.xml file. Parameters such as text color, size and style, shadows, background and gravity are grouped and used to improve the look of the project UIs.

Figure 9 shows the source code for a Style named text_small. In this style, the text color is black, the text font size is 12 sp, and there is a shadow effect around the text.
Figure 10 shows another example of a simple style, called small_text_white_backgroud that implement a custom background similar to a white box. This particular style is used on some of the textviews in the main.xml interface.

Conclusion:

The BSU_Campus_App is an Android application that provides users with information about the BSU Campus on their phone.

When users are in the vicinity of each of the BSU buildings, the UI correspondent to that building is displayed on their device providing them with information about that specific building. This information includes the building’s name and picture and a description of
the public spaces inside that building such as offices, classrooms, auditoriums, etc. Each one of the BSU Campus buildings has its own UI.

This App also provides Users with real time access to important Campus features that includes Student Announcements, Events Calendar, Class Cancelations, Colleges’ web pages, InfoBear and Student and Employee email.

Users also have access to a list of the BSU Campus buildings from where they can have access to information related to each BSU building without being on campus.

**Future Work:**

This section provides some examples of features that can be implemented in the future in order to improve the Bsu_Campus_App:

- **Announcements Portal:** This feature will allow users to create and share announcements about their events on campus. Alerts about coming events will be sent to the phones of the Bsu_Campus_App users.

  This feature will be formed by:
  - A Database that will store information about places in the BSU Campus.
  - A Android Activity that will use the JSON format to display the data read from the database.
  - A php script that will be responsible to read the SQL data from the database, convert it into the JSON format and send it to the Android Activity.
  - A script that will validate the users to ensure that only members of the BSU community are allowed to create and share events.
- **Bus track:** This feature will provide the real time location of the BSU Campus Buses. This feature can be done by integrating in this App another Application which is being implemented here at BSU by two Computer Science Graduate Students advised by Dr. Uma Shama.

This integration will require that all the BSU Campus buses must be equipped with a device able to store information about its location in terms of Latitude and Longitude in a database.

It will be formed by:

- A php script to read location data from the database and send data to an Android Activity.

- An Android Activity that will display images of the buses in the BSU Campus path. The images to be displayed will be selected in dependence of the location coordinates received from the devices in the buses.
References:


