

2013

Blackboard on Android: a Free and GVSU Centric Blackboard Mobile App

Jacob V. Romero
Grand Valley State University

Follow this and additional works at: <https://scholarworks.gvsu.edu/cistechlib>

ScholarWorks Citation

Romero, Jacob V., "Blackboard on Android: a Free and GVSU Centric Blackboard Mobile App" (2013).
Technical Library. 165.
<https://scholarworks.gvsu.edu/cistechlib/165>

This Project is brought to you for free and open access by the School of Computing and Information Systems at ScholarWorks@GVSU. It has been accepted for inclusion in Technical Library by an authorized administrator of ScholarWorks@GVSU. For more information, please contact scholarworks@gvsu.edu.

Blackboard on Android: a Free and GVSU
Centric Blackboard Mobile App

By
Jacob V. Romero
December 2013

Blackboard on Android: a Free and GVSU Centric Blackboard Mobile App

By
Jacob V. Romero

A project submitted in partial fulfillment of the requirements for the degree of
Master of Science in
Computer Information Systems

at
Grand Valley State University
December 2013

Jonathan Engelsma

Date

Table of Contents

| | |
|---|-----------|
| Abstract..... | 4 |
| Introduction..... | 4 |
| Background and Related Work..... | 5 |
| Program Requirements | 7 |
| Implementation | 9 |
| Results, Evaluation, and Reflection..... | 14 |
| Conclusions and Future Work..... | 15 |
| Bibliography | 16 |

Abstract

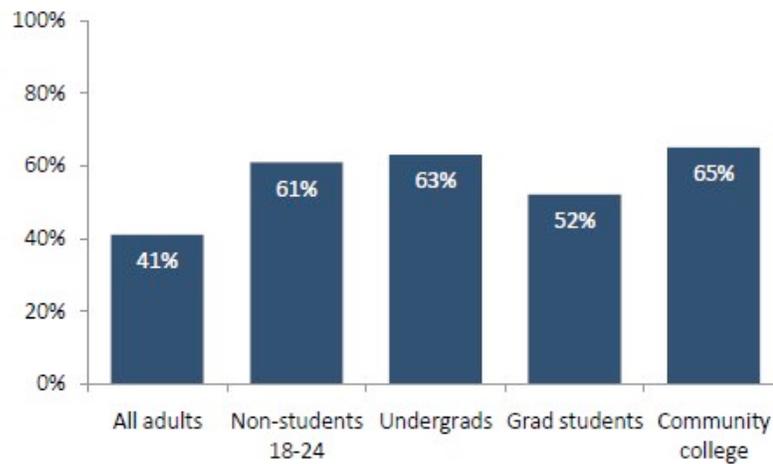
Blackboard Learn is the current IT supported Learning Management System at Grand Valley State University. This system has gained popularity at GVSU and much of the faculty use it for their courses, and therefore students use Blackboard for their course work. In the fall of 2013 over 95% of students have at least one class that is using this system, and because of the widespread use many students would like to access Blackboard with their mobile device. Currently GVSU students can check their class schedule, gather current news, track the location of a bus, look at their final grades, and much more from free mobile apps developed at GVSU. However there is no free app for student to access their courses on blackboard. This project will develop a free Android app for GVSU blackboard. This project will require a backend Java application (Blackboard Building Block) and an Android app. The features of the application will allow student to view their most recent announcements, their next due assignments, and any grades for their currently available assignments. It will be free and branded for GVSU. Finally the application will be able to be changed and adapted to fit the growing mobile needs of the University.

Introduction

Grand Valley State University has been using Blackboard for its course management system for several years. Although the amount of use started low, it has grown significantly in the last few years. If we look at it from a student perspective, in the fall semester of 2013 over 95% of all students have at least one of their courses available on Blackboard. In 2010 the PEW group found that 63% of college undergrads were using a smartphone to browse the Internet or check their email (see Figure 1) (Aaron Smith). When reviewing these numbers it has become apparent that more GVSU services should be accessible on mobile devices.

Cell phones and internet connectivity

Percentage of cell phone owners in each group who use the internet or email on their cell phones



Source: Pew Research Center's Internet & American Life Project 2010 tracking surveys. All include landline and cell phone interviews. N for all adults=9,769; n for 18-24 year old non-students=717; n for four-year undergrads=246, n for grad students=112, n for community college students=164.

Figure 1 – Pew 2010 Survey

The motivation to allow easy access to GVSU services through mobile devices is evident in the numbers, but also Blackboard services specifically have been requested by GVSU students many times. Both GVSU IT and the GVSU Computer Information Systems Mobile Application and Service Laboratory (MASL) group have developed free mobile apps for the GVSU community. A often requested features of both of these groups is the ability to access Blackboard course data and specifically important to students, grades.

The Android application produced for this project (myBlackboard) will allow students who have these mobile devices to quickly view some of the most recent information in their current blackboard courses. They can see available announcements, items with due dates, and grades for each of their courses. The myBlackboard app will be free to download and use by the entire GVSU community. The main goal of this project is to create a working Android application that is published in the Google Play Store. Along with this goal this project requires the development of a backend on the Blackboard server to allow a connection to Blackboard data for myBlackboard and future mobile applications.

Background and Related Work

At the time of writing this paper there is only one mobile application available for Blackboard Learn. This app is available on Android, iOS, and Blackberry. The app is free to download, however there is a license fee to access the content on the user's blackboard site. The license fee is either \$1.99 for a year or \$5.99

for lifetime. One excellent feature of this app is the design, which includes the ability for a user to customize the colors for each course. In Figure 2 and Figure 3 you can see the simple and sleek design of the Blackboard application. Also the user can customize the application by assigning different colors to each course. This provides a quick visual representation of what class a content item is associated with. Another way this application excels is in its larger feature set. Students can easily get any content that is in their blackboard course. Some examples are grades, announcements, content items, assignments, tests, blogs, journals, and discussion boards. Also a user can setup any changes to their course to be sent to their mobile device as a notification. Not only can users view their content but also they can add and interact with it. Faculty can add announcements, content items, and comments and students are able to post to discussion boards, journals, and wiki's for example.

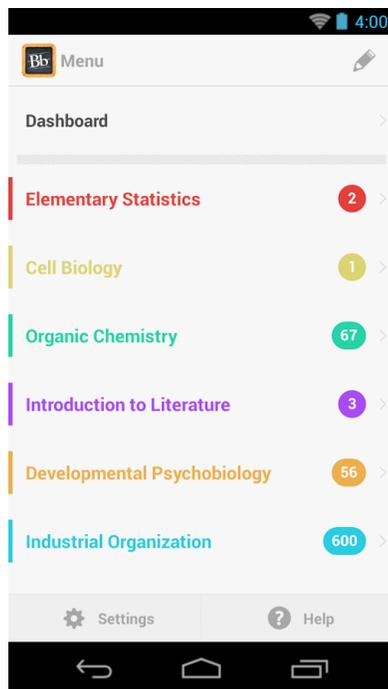


Figure 2 - Blackboard App Course List

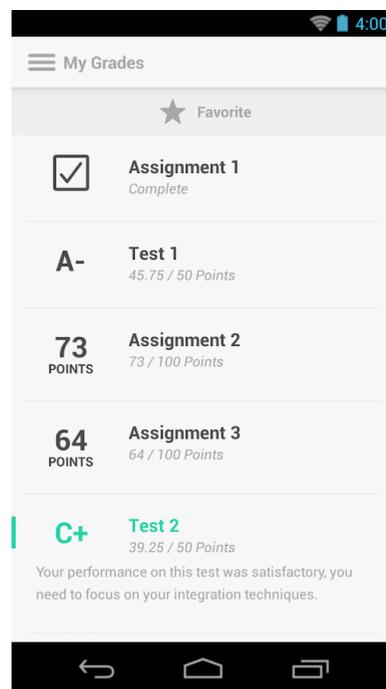


Figure 3 - Blackboard App Grades

Unfortunately the bugs and slowness of the application far out weight the design of the app. This app has poor ratings, with over 15,000 user ratings over 7,000 of them are one star. Many of the poor ratings describe not being able to use the app because of errors and bugs.

Finally the Blackboard app does the backend well. Blackboard Inc. has designed a building block that comes pre-installed in any new install or upgrade of the Blackboard product. This building block does not require any more actions from the University. It is simply there, ready to go, and fully supported by Blackboard.

Although there is only one mobile application for Blackboard, there are some mobile applications for Moodle, which is another learning management system. In this app the design, although not special, is simple and easy to use (see Figure 4 & Figure 5).

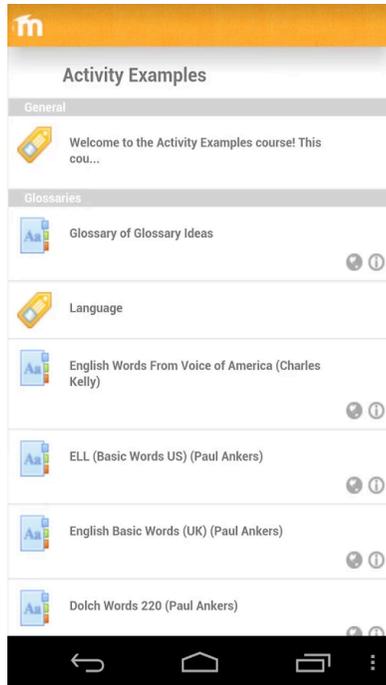


Figure 4 - Moodle App Content



Figure 5 - Moodle App Dashboard

The list of features is also simple but it does the job of giving the user access to view data in their courses on a mobile device. Although the feature list is much smaller when compared to the Blackboard app, this app is free for anyone to download and use. So it is much more accessible for college students, who do not want to add one more fee to their course work.

Similarly to the Blackboard mobile app, the user ratings are bad for this app. Out of 182 reviews 93 are one star reviews and many users complained of the app not working or being slow and jumpy.

The backend is not as simple as the blackboard app, and it requires a specific setup of the Moodle server. If a University is using Moodle in a different way this app will not work. The extra work for a University's IT department may discourage possible adoption.

Program Requirements

Blackboard Learn has a large list of features and items that can be used in a course. Each faculty member decides how and what to use in their own course. So, the first requirement that was needed was to decide which set of features the app would be able to support. Then out of those, to decide the ones that made the

most sense for a mobile device. When considering a mobile device it was expected that most mobile users would want recent and timely information. For instance a student could be looking for their grade of the test they took yesterday, or the student would like to be able to quickly see the new assignment that was posted today. With this in mind it was decided to include a small set of specific items, which are:

Announcements – Course announcements posted by the instructor.

Grades – Any column in the grade center that is visible to the student.

Due Items – Any item that has a grade associated with it.

Send Email – Send email to faculty of a course.

With the knowledge of the content types, the next step was to understand the requirements of the design of the app. One concern was the number of items to list of each content type and how much detail to display for each. Again, Blackboard is very customizable and each course is likely to be very different. One instructor may heavily use the announcement feature or may have several small assignments listed every week, while another may use only one or two items the entire semester. This variation in course content led to the next requirement of a dashboard list view and filters on the content. The dashboard list view would display the currently request list based on the filters selected. There would need to be a limit on the number of items displayed on a page for each content type. Also a user should be able to select a course, a content type, or both to filter the list. When the user selected an active filter it would change the number of items they received depending on what was selected. Eventually, a table was created to describe how many items would be displayed based on the current filters set. (see Figure 6).

| | All Content | Announcements | Due Items | Grades |
|--------------------|---|---|---|--|
| All courses | Ann - 5 most recent Due - 5 next Due Grades - 5 Most recent graded items | Ann - 20 most recent across all courses | 5 next due across all courses | 10 most recent with grades across all courses |
| One Course | Ann - 10 most recent for selected course Due - 10 next Due for selected course Grades - 10 Most recent graded items | Ann - all Announcements (limit 50) order by most recent | All Due Items that have not been submitted oldest due first | All grade items with and without grades ordered by relative date |

Figure 6 - Content Based on Filter

Once it was decided how many of each item for each screen, the next step was to decide how to filter each item type. The goal was to determine what the user would want to on top of the list, and this would be the content that is most import to the user typically. Each item type did present its own challenges.

Announcements were the simplest. An Announcement in Blackboard is either available because of dates set or it is always available. We cannot just simply use the date the announcement was created because it may have been created in advance and the dates set for later in the semester. If the Announcement was date

restricted, then the start date was used for the sort date. Otherwise, the created date was used. The list of Announcements was then sorted by most recent.

Next was the grades content type. After the available list of grades is collected we need to decide which date to use as the sort date. It was decided that if the item has been graded then the date the grade was posted is used. Otherwise the date the item was last modified was used. Then the grades were sorted first by most recently graded item, and then by most recently modified.

Lastly, due items need to be filtered and ordered. The list of items is created from the list of Due Items in which the user has not completed, which is the list items in the course still due for the user. Then the sort date is based first on the item's due date, then if there is no due date it is based on the end date of the item, and if there is no end date the last modified date is used. The entire list will be sorted by the oldest first. The order is reversed for these because the user will look at what is past due first, then at the next due items.

With announcements and due items, there is a description and can be attachments. The application will need to retrieve and display this information. When in the main list view a plain text preview of the item's description will be displayed. Also if the item has attachments an icon is displayed on the list item view.

The last piece of the requirements is what information to get from the Blackboard server and what to store on the device. Currently the myBlackboard app is only retrieving content from the Blackboard server and is not pushing any data. The data types the application will be requesting are authentication verification, course list, faculty list for a course, list of announcements, list of due items, and a list of grades. The announcement and due item types will require a preview of the description as well. This preview will include stripping out the HTML content of the description so a plain text string can be added to the list view item. Next the backend will need to have a way to load the full HTML descriptions for both announcement and due item types individually.

Implementation

Platform

The first consideration was what mobile OS would be used. Ideally both iOS and Android would be developed in order to give the most amount of access to the GVSU community. However, due to time constraints and the back end requirements, Android was chosen. This was the preference because of prior experience and familiarity with both Java and the Android development environment.

A Blackboard building block is the only choice for the backend of this application. Building blocks are Java web applications that allow a developer to connect into the Blackboard Learn environment and add new features. They are Blackboard's version of a plugin. This project required the use of a building block because only a building block can connect to all the Blackboard APIs and can connect to the Blackboard database. Both of these items were needed to retrieve the data necessary for the myBlackboard app.

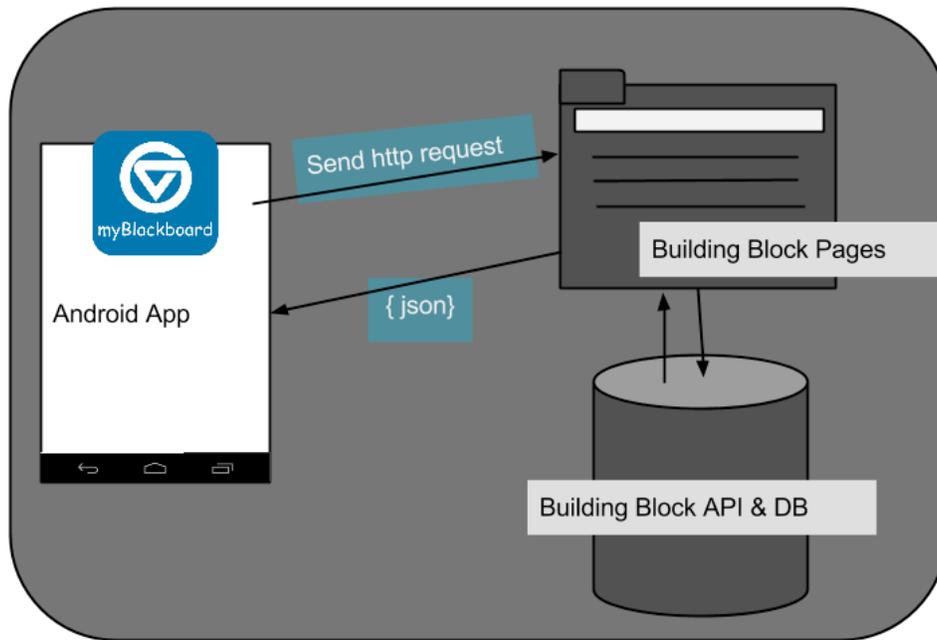
Retrieving the Data

The Android app does not contain data other than the username and password of the user. All the actual content comes from the Blackboard web application. Which is why the building block will have external web pages that will return data based on the passed parameters.

Step one for the application is to authenticate the user with Blackboard. Blackboard is a web application and using the browser the user would normally authenticate over an https connection. The myBlackboard app does this just the same. It uses an Android HttpClient to send the same request to the Blackboard server as if it were a normal browser session. At this point it is crucial the myBlackboard app stores the session cookies. Keeping the session cookies and making them accessible to all the activities in the app, allows the Blackboard server to verify user authentication and identity, and the data cannot be accessed without it. Using the CookieStore class and attaching it to the HttpClient request, all cookies that are returned in the header are then accessible upon completion of the request. In order to store the cookies across all the activities in the class, the CookieSyncManager & CookieManager classes were used. After an HttpClient request completed all cookies in the header were retrieved from the CookieStore and added to the CookieManager, then a sync was forced. Forcing the sync is necessary because if there was a delay, the next request might not have the necessary cookies associated with it.

Once the user is authenticated, the data is then requested. Looking at the requirements we know that the myBlackboard app will need to be able to retrieve a course list, all available announcements, all available grades, all available due items, and a instructor list for every available course. Also the application needed to be able to check to make sure the session cookie was still active with Blackboard, which verifies authentication along the way. Finally, for each of the content items (announcements, grades, and due items) a limit and course filter could be used.

As a high level overview, Figure 6 shows how the data is retrieved. The myBlackboard app sends a request to an external building block page. The page will import the appropriate classes developed for the building block. These classes act as a wrapper around the Blackboard API and provide specific helper functions. The building block classes return the requested data, and the page outputs a JSON representation of the data. Then, the myBlackboard app pulls in the JSON string and displays it properly based on the current situation.



(Aaron Smith)Figure 7 - Architectural Diagram of Data Retrieval

Before describing each building block external page, it is important to understand what each of these has in common. Each page will be importing the necessary Building Block helper classes. These classes will have the functions which access the Blackboard API or query the database and return a list of only what the external page requested. This extrapolation adds another layer and allows one control point for each type of content the building block will need. It also simplifies the external pages code, and makes expanding the options much easier. From a maintenance standpoint, Blackboard is typically updated at least once a year and this will allow one point to update if the API changes.

These building block classes are broken down into a class for each content type. There is a class for announcements, grades, content (which includes due items), courses, and authentication. Any item that needs to be sorted also has a comparison class included in the building block. These classes implement the Comparator Java class, which is used to sort a list of items, depending on the data requested. For instance grade items have their own Comparator, which allows the sort to give priority to a grade item that has been graded.

The authentication class is another class used by each external page. Each page must first create the SessionManger class using the http request sent to the page. Then it can use the isAuthenticated function to be sure the user is authenticated. In some cases the page might also use the SessionManger class to verify the user has access to a course, announcement or content item. Adding these functions in this class allows a quick way to be sure no data is being accessed, which should not be.

Building Block External Pages

Course List Request

This page will be used to retrieve the list of currently available courses in the Blackboard system for the specified user. The process is to verify the authentication and username, which is accomplished by the SessionManger class. After that the building block Courses class is used to retrieve the list of available courses and the page outputs the results into JSON.

Dashboard Request

The dashboard request page allows the myBlackboard app to retrieve a list of all three content types in one request. This page is used when the app has no content type filter set. Once authentication is verified the page must look for the parameters of a course id (crs_id) and limit included in the request. If there is no limit then the default limit is used. When there is no course id value data for all the users' available courses is fetched. Each of the content type classes are imported, and used to retrieve the data requested. The limit and course id variables are passed to these functions and the building block classes will only return the appropriate content. Lastly the JSON representation of each of these lists of content items is outputted to the response.

Announcement, Grade, and Due Item Requests

These are three separate pages. Each page is similar to the dashboard page in process. Each can receive a course and limit variable and each uses the building block classes. The values returned from the building block classes are sent to output in JSON.

Course Instructor List Request

This external page will output instructor information for each of the user's available courses. The process is similar, and after the authentication is verified a list of instructors for the course passed is retrieved from the building block classes' function. A value for crs_id must be present in the request. Then the results are sent to the output in a JSON format.

Detail List

Two separate external pages accomplish a request for the description of either an announcement or due item. Each page will connect to the corresponding building block classes and return only the details of the item in HTML. Thanks to the Android WebView this is a simple way to display the details of these items.

Displaying the Data

Overview of myBlackboard

Before going on too far we should understand how the application flows. A user logs into blackboard (see Figure 8). Once authenticated, the dashboard view is assumed, and a list of all content types for all available courses is fetched. From here the user can select a content type or course from the navigation (see Figure 9), which changes the filter and triggers a new fetch of data and an update of the list. The filters are saved in the SharedPreferences of the myBlackboard app, which allows easy access. This means the myBlackboard app only needs one function to trigger a fetch when filters are adjusted. Also the user can select a due item or announcement to see the full view, and the details are retrieved from the building block and displayed directly in a WebView.

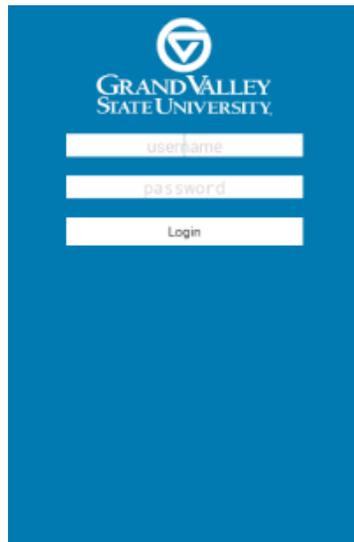


Figure 8 - Login Screen Shot

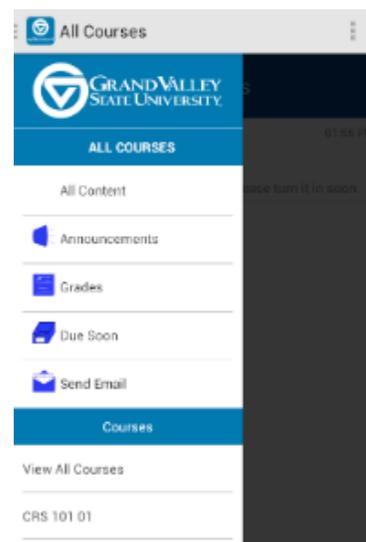


Figure 9 - Navigation Screen Shot

Classing the Requests in Android

With all these pages and parameters for each request it made sense to setup a similar class structure in the myBlackboard app as in the building block classes. Each type of request that will need to be made has a set of classes in the myBlackboard application. The first class is a fetch class that organizes the requests, kicks off the AsyncTask and is responsible for ingesting the data returned in the response. Second, there is a class that holds a content type, and contains the needed variables needed for each content type. Then in the fetch class it can hold a list of the content type for the most recent fetch.

When the user selects an option, the new filters are saved to SharedPreferences and then the *fetchForPrefs()* function is kicked off. This function will gather the filter options and pull in the correct fetch class, and then kick off the request using the appropriate function in the fetch class. The fetch class sends a request to the correct building block page, and when it completes it takes the JSON string and

processes it into a list of the content class items. Finally the list activity is notified of the change to the list, which is when the list view is updated.

Now the views need to be created to display the data. The main list activity uses a custom ListAdapter that extends Android's BaseAdapter. This iterates over each of the content lists from the previous fetch and creates a view for each, inserting it into the list view.

Figure 10 provides the sequence of events from the users click to the view being update in the myBlackboard app.

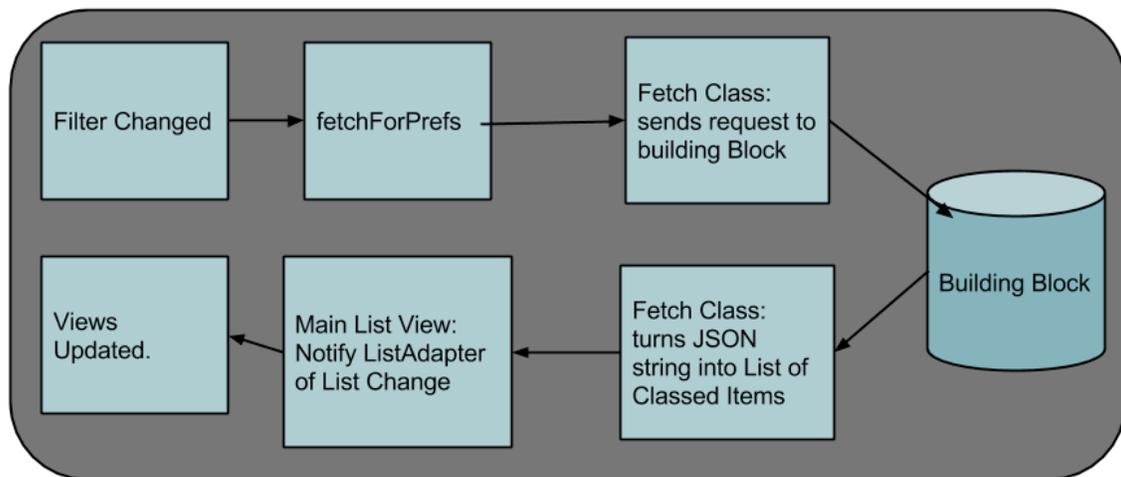


Figure 10 - myBlackboard Fetch Process

Results, Evaluation, and Reflection

At this point the results and the evaluation of the myBlackboard application are difficult, because it has only been available in the Play Store for about one day. Initially the app was posted to an external web page and information was released to a small number of GVSU courses. A survey was setup to retrieve any responses to the application. Only two responses came back during the 10-day period it was available. Both of these results gave the application 4 out of 5 in ease of use and both said they would use it in their course work at GVSU.

Also during that time frame there was some usage data gathered from Google analytics. Total there were 22 new users and 97 total sessions. Also, there were four unique crashes and all were fixed before release into the play store. I would say that although the numbers were not huge the results were effective, for a small “beta” test.

The final product does complete the goals of this project. First, it gives the GVSU community a free mobile application that can be used to access Blackboard content. The content makes sense for mobile devices and if nothing else, allows students a way to check their blackboard course grades on their mobile phones. Secondly, this project created the framework for future mobile development in blackboard, by creating a backend through a building block. The addition of this building block to GVSU's Blackboard instance will allow for other application to be developed as the need arises.

Conclusions and Future Work

First, mobile development is very appealing. The mobile phones and tablets are exciting devices. The new screen size, the instant access and the constant access all give new perspectives to any application.

This application has been published to the app store and should prove useful to the GVSU community. Overall developing the application was a great learning experience and debating and deciding how best to simplify Blackboard into pocketsize devices proved to be challenging, but in the end achievable.

The very next step for this project will be to create an iOS version of myBlackboard. Only having an Android application will be an issue for many in the GVSU community. After an iOS version is completed, a few small enhancements could prove very useful. Some examples are:

- Faculty can add announcements
- Faculty can comment on a grade
- Students can view comments of a grade
- Discussion board support

However, before adding too many new features, there needs to be a testing structure in place. This will allow future upgrades of the Blackboard environment at GVSU to be less hindered by the custom building block and the myBlackboard applications.

Finally, an excellent addition to this project would be to share. The building block could be packaged and shared with other Universities using Blackboard. Then create a library application for Android that can be pulled into any new Android application. This would allow other universities to be up and running with a free mobile version of Blackboard with very little additional work. Also I think all kids should share their building blocks.

Bibliography

Aaron Smith, Lee Rainie, Kathryn Zickuhr. College students and technology. 19 July 2011. 01 12 2013
<<http://www.pewinternet.org/Reports/2011/College-students-and-technology/Report.aspx?view=all>>.