**FSC Mocs App**

Author

Tyler Jackimowicz

Abstract

Florida Southern College students are all directed to use the portal or home page to locate necessary information. This poses issues with time efficiency and extensive searches. FSC Mocs App consists of an application that could possibly be utilized in the near future and would provide another option for the student body. I utilized a free web hosting server, 000webhost.com, to contain a SQL database, using phpMyAdmin, to be accessed by an Android application online using Android Studio. This application is based on Florida Southern College student information that would be pertinent for students to access. Portal has a profuse amount of data that may not be necessary to display when searching for particular information. Students, in general, would prefer only the basic information presented to them without the hassle of searching. This is where my simple phone application comes in for the student to log in and access their information.

Keywords

Android Application, MySQL, Web Server, Database, phpMyAdmin, Android Studio, Java, XML, PHP.

Introduction

Programming one through three, database management, and web development classes provided me the necessary skills to complete this project. Working as a Mobile Game Developer intern also provided me with insight to design this application. With this previous experience, I was able to work with the Java and XML coding, web hosting server and MySQL database without trouble and understood the process of these. There is beneficial documentation provided online that has assisted with this project in order to combine the user interface (UI), database information, and mechanics of the application to work cohesively. Once all of these operations were connected, the FSC Mocs App was ready to be tested.

Problem Statement

Overloading students with information on a single website has been a continuous issue with Florida Southern College’s portal. In order to find specific information you are clicking and searching multiple times to be able to access it. This problem poses as a weakness in regards to time efficiency and it further complicates the process. Another issue with the portal is overloading the server with too many users at one time, causing it to crash or run extremely slow.

My proposed application would give the students an option on how they want to retrieve their information. This would prevent wasted time and frustration when trying to find simple data quickly and effectively. It would also redirect traffic from the website, causing an equal distribution of users.

Review of literature

**Research 1**

After searching for related literature, I found a research paper on efficiently and effectively detecting mobile app bugs with AppDoctor by Hu, Yuan, Tang, and Yang (2014). This team of researchers from Columbia University designed a solution to app debugging, AppDoctor. This application provides a system for testing apps against many system and user actions, and helps developers diagnose the resultant bug reports. It is implemented in Android and operates as a cloud of physical devices or emulators to scale up testing. It has evaluated 53 out of 100 most popular apps in Google Play store and 11 of the most popular open-source apps. The application uses approximate execution and action slicing to speed up debugging and find false positives and bugs for the developer. Once this program was used on those most popular applications, it detected 72 bugs, verifying 43 bugs and 16 false positives automatically and required only 5 reports for developer inspection. I found this research to be useful in understanding the necessary process of running a phone application and how to handle debugging. This would be an extremely useful tool to use in the future.

**Research 2**

The next research topic I retrieved was regarding software engineering issues for mobile application development by Wasserman (2010). Wasserman provides an overview of important software engineering research issues related to the development of applications that run on mobile devices. Among the topics are development processes, tools, user interface design, application portability, quality, and security. The literature provided insight on the numerous programming environments for mobile platforms such as the Xcode package for Apple IOS, Android Development tools plug-in for the Eclipse programming, and Visual Studio environment for Windows applications. Developers designing small to medium-sized apps are encouraged to adhere to the important principles of abstraction and modularity that are built into the platform architectures. As the apps become more complex, it will be essential to apply software engineering processes to assure the development of secure, high-quality mobile applications. An important development technique used universally to create applications is called Agile Scrum, a sequence of 2-4 week “sprints” where a team addresses a set of tasks for each individual to complete.

Another key aspect to consider is that gestures, sensors, and location data may be used in game consoles and traditional computers, however, they play a dominant role in many mobile applications. The mobile user interface paradigm is based around widgets, touch, physical motion, and keyboards rather than the familiar Windows, Icons, Menus, Pointer.

Wasserman (2010) defines software engineering as a process by which an individual or team organizes and manages the creation of a software-intensive system, from concept through one or more formal releases. A question he addressed is how software engineering relates to application development. Software engineering and developing apps both face common issues. These include integration with device hardware, issues of security, performance, reliability, and storage limitations. Mobile applications also present some additional requirements. These include potential interaction with other applications, sensor handling, utilization of networks and the internet, supporting different operating systems and devices, security, user interfaces, testing and power consumption.

For mobile devices and their apps, the software engineering process must not only be aware of the hardware properties, but also the project management issues and unique aspects of mobile applications that I previously mentioned. It’s in the interest of the application developer to adhere to platform standards, especially on touch-screen devices, where users expect to use the platform’s standard set of gestures, which differs for each platform. You must also consider what functions should be present in the app and how various forms of input and sensor data will be utilized. Other issues most relevant to mobile applications are performance, reliability, quality, and security.

Testing is another important area for mobile software engineering research. It’s insufficient to only test an Android application on an emulator; it must be tested across many different Android devices running different versions of the operating system. Another area to consider is app maintenance. While “early-adopter” consumers are often willing to update their device and their applications, most enterprise users are less likely to do so.

The next essential task to consider is from the standpoint of the application developer, it’s quite expensive to support multiple platforms, especially when there are multiple versions and variants for each of them. The application developer has several options: Develop for a single platform only and use a common feature available across all variants of the device, Develop native applications for each platform and variant and trade off development costs against ability to optimize app for each device, or Develop mobile web apps, thus minimizing amount of native code for each platform.

While the large number of mobile applications makes it appear that software development processes for them are well understood, there remain a large number of complex issues where further work is needed. In addition, there is a mobile “angle” to almost every aspect of software engineering research, where the characteristics of mobile applications and their operating environments present a new or different set of research issues

**Research 3**

Another advantageous research article I found was regarding a mobile food ordering application using Android OS Platform by Michael Yosep Ricky (2014). Mr. Ricky’s research was conducted in order to design an Android application to allow users the ability to order food. This application allows a new order, order history, restaurant profile, order status, tracking order, and setting profile features. Ricky viewed an increase in population in Indonesia and the growing number of restaurants as a way to help others by creating this phone application. This application allows more than the competitor’s ordering services, such as websites, because it provides additional features. The user would have the ability to view their order history, such as the last order all the way up to the last seven orders. The user also has the ability to view restaurant’s profiles to find information regarding the restaurant and make a call using this feature. The order status tracks the user’s request and shows updated messages, such as order received, order confirmed, cooking, etc. The other features include menus, GPS tracking, lists of restaurants, and making an order on the application.

Ricky evaluates his application using eight golden rules. The first rule is consistency, in regards with using the same font color, layout, menu, and menu icon. The application must provide universal usability meaning that it is user friendly on both customer and courier side. Another key rule is that the application provides information feedback to users if an error occurred and that it shows a dialog. The application shows a final dialog that provides an understanding of what actions the user is taking. Ricky designed a simple error handling process to avoid fatal errors. Another rule is that there is a reversal action allowance, such as always having a back button in the application. The next rule is that the user becomes the controller and the system will respond to the user’s action. Lastly the application reduces the information capacity for short-term memory. For example, the app uses relevant icons with menus such as the exit button having an exit door icon.

This application provides useful tools for the users and restaurants and offers an innovative way to order food. These eight golden rules provide a great insight on how to design a productive application. I learned a great deal about what should be focused on when designing applications and how to maintain the application.

**Research 4**

Lastly, I found an article regarding Android applications security by Pocatilu (2011). Worldwide, one million people were infected by mobile malware in the first semester of 2011. It is stated that only 4% of smart phones and tablets use antimalware and antivirus software. These malware applications can be used by attackers to send and/or receive confidential data, malicious applications and to make unauthorized actions such as using Bluetooth to connect to other devices, telephone to make unauthorized calls or recordings, messages to send confidential content to attacker, wireless networks to connect to the internet to transfer data, and near field communication to make unauthorized payments.

Behavior of malicious applications could vary from annoying messages to very unrecoverable damages. These include: unwanted web pages opened, advertising popup, unauthorized use of personal information data, confidential data transferred on a remote location, altering data stored in file system, contacts, messages, etc. Depending on the platform and/or operating system, applications can: read and write contacts, read and write calendar entries, read short messages, send emails, make phone calls, get location, access the internet, and read and write on file system.

Android operating system is based on Linux kernel so that applications isolation, system and security rules are Linux specific. Android application files are packed in files with apk extension. These files contain all the classes and resources required by the application. Each application runs in a separate virtual machine, having its own unique identifier. Due to this, application resources are protected from other applications and the communication and data transfer between applications has a high degree of confidence and security. Even so, applications could communicate with each other using messages, this being another source of threat.

Another aspect to consider is that Android applications can be installed from various sources, including: Android market, alternative online shops, own developed applications, third-party developers, and unauthorized sites. Applications published on Android Market by the developers do not need approval from Google like applications published on the App store (apple) or Marketplace (Microsoft).

Application permissions allow the application to utilize the user’s phone features. Depending on what kind of permission, certain features can be utilized. Permissions can fall into a normal or danger category. Normal category is where the features used don’t present any risk for application or system and the user is not informed when application is installed. Dangerous category is when the application has capabilities that, if used by malicious code, could produce negative effects and the user must be aware of these. All android applications need users’ permission in order to access potentially dangerous features. The Android security model is based on these permissions that the users have to be aware in order to prevent malicious applications installation. A serious security issue is that attackers will write their own applications or modify existing ones by reverse engineering. For example, existing application published on Android market could be modified with malicious code inserted and published with a different name.

Malicious applications can utilize services very easily if coded right. They can install a Broadcast Receiver for incoming messages in order to get access to private information. As the messages arrive, they are received by the malicious application and processed.

Lastly, there are prevention measures that need to be taken when considering phone applications. Installation from trusted sources and based on other users reviews and scores is very important when installing applications. If there are complaints about spyware or malware built in the application, the user will be informed. Users should read carefully the application permissions and if they have any doubts regarding the application intentions they should cancel the installation immediately.

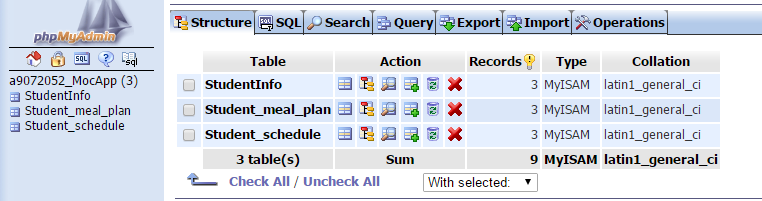
Another option is to install applications that monitor the system for malicious code preventing the infections. When creating an application and securing the user’s information, one solution would be to encrypt the users’ sensitive data. Another kind of system can encrypt and decrypt data on request or they can monitor services and activities and make these actions on the fly, as data arrives. One way of doing this is having received data encrypted and stored in a database or on external storage. Some of the intercepted messages are not delivered to default applications in order to avoid writing plain data on the file system or databases. When data is required by an application, it is extracted from the SQLite databases, decrypted, and sent to those applications.

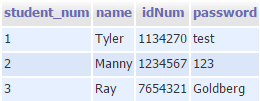
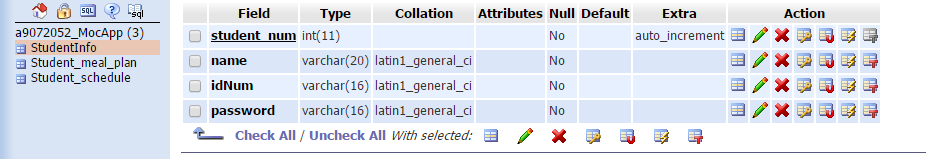
Mobile application security is very important today due to increasing number of users and the importance of personal and confidential data stored on mobile devices. In order to help secure users’ data, it is important to encrypt and decrypt all private data as they are accessed in order to assure its confidentiality. I took this perspective into account and in the future, it would be important to secure user’s data and utilize advanced security features.

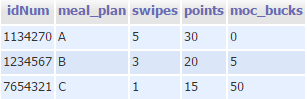
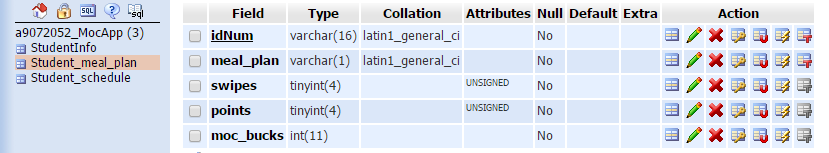
Work to date

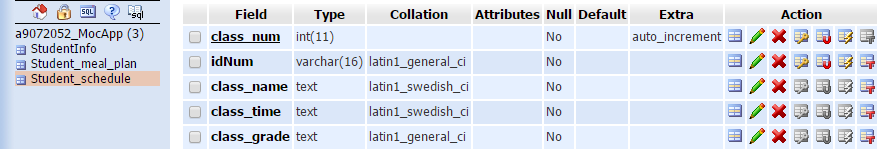
The FSC Mocs App is essentially complete. There is, however, further room for improvement in the future. Currently, the SQL database is filled with test data for the Student, their meal plan, and class schedule. This was completed using phpMyAdmin to create a SQL database on the free web hosting site, [www.members.000.webhost.com](http://www.members.000.webhost.com). The UI is complete and was designed on Android Studio using XML coding and the drop-down menu provided in this tool. The mechanics of the application is complete and was designed using Java coding. There are a series of Java classes that work cohesively to log the user in, retrieve information from database, store objects and display necessary data in the UI. The Server Requests java class calls PHP files located in the web hosting server file manager in order to receive the information from the database and stores it as JSON objects.

Database Tables



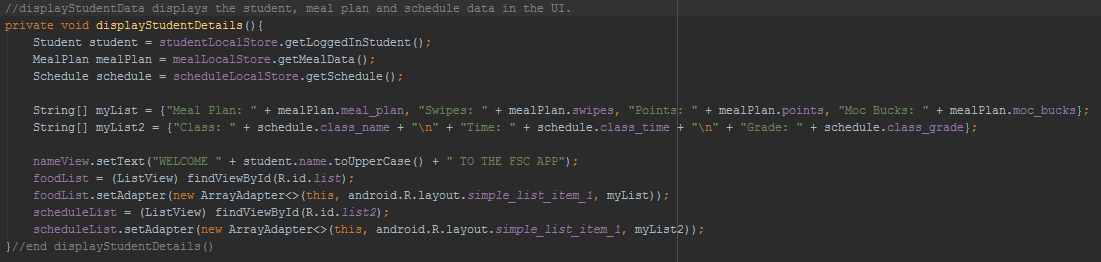
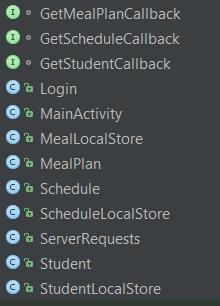


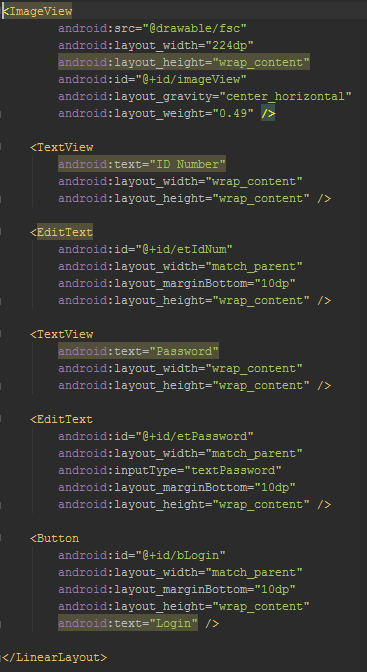
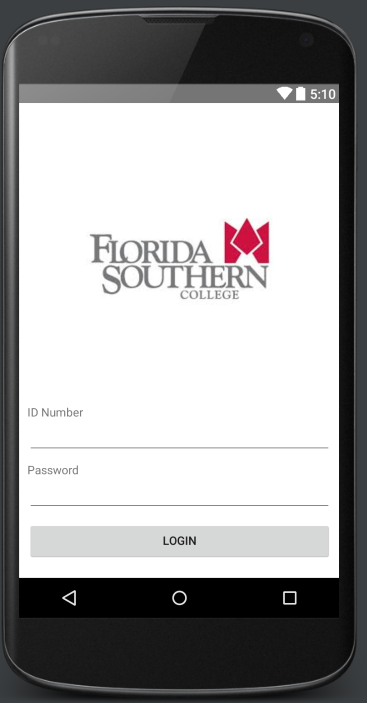






Java and XML class examples





Analysis

FSC Mocs App would provide another useful resource for the Florida Southern student body. This application would cut down traffic for the website, allow for quick information retrieval, and result in overall satisfaction. I faced issues along the way when designing the java classes to connect with the database. After researching and trial and error, I was finally able to get the database to be connected and accessed the information. Once connected to the student information table, I had to connect the application to the schedule table and the meal plan table. This took some time to alter the code and utilize the authentication in order to make sure it only retrieved the data that matched with the ID number. The only unresolved problem was trying to retrieve multiple class objects at one time from the database. In real application, this would pose a problem due to students having multiple classes. At this time only one class object can be retrieved. All of the functions work without issues, however, and this problem can be a future task to complete.

Conclusion

This project has provided me the opportunity to see the step by step process of creating a user interface, database, and the mechanics to run an application. This process is conducted on a much larger scale for major companies in order to create applications. I plan to use the knowledge I gained from this project to help with career searching and design applications of my own in the near future.

Further Research

In the future, FSC Mocs App could be enhanced immensely. The classes display tab would be fixed in order to store and hold multiple class objects for each student’s schedule. In regards to the database, the data would need to be secured using advanced encryption instead of plain text. There are also multiple features that could be added to this application. For example, I would add the interactive campus map, updated calendar to provide students insight of upcoming events, and updated menus for the multiple food locations provided on campus. These additions would increase the usefulness of the application tremendously.

References

Hu, G., Yuan, X., Tang, Y., Yang, J. (2014). Efficiently, effectively detecting mobile app bugs with AppDoctor. *EuroSys ‘14 Proceedings of the Ninth European Conference on Computer Systems*, 18, 1-15.

Pocatilu, P. (2011). Android Applications Security. *Informatica Economica*, *15*(3), 163-171.

Ricky, M. Y. (2014). Mobile Food Ordering Application using Android OS Platform. *EPJ Web Of Conferences, 68*(5).

Wasserman, A. (2010). Software engineering issues for mobile application development. *FoSER ’10 Proceedings of the FSE/SDP Workshop On Future of Software Engineering Research*, n/a, 397-400.