Cheyenne Heckman

Prof Brooke

[CS-499-T5648 Computer Science Capstone 22EW5](https://learn.snhu.edu/d2l/home/1071635)

05 June 2022

**Professional Assessment**

**Introduce**

One artifact was chosen to represent growth in software engineering, databases, and algorithms and structures. That artifact is the android app I had developed for an application course. This artifact was chosen in particular because apps typically incorporate a variety of different elements. Databases to view information that the user interacts with, software complexity which could include the display, or the structure of the app. Which is important to ensure security and decrease the amount of software patches that could be needed.

**Development of this App**

Purpose: The purpose of this app was to act as an event viewer similar to apps such as EventBrite and Ticketmaster. The user is able to view event information (such as event title, event location, and event time.) and add it into the favorites page.

**Previous Version**

Before the event app was updated with new information. The app was fairly simplistic due to limited knowledge of java and databases, but even though it was fairly simple it held a lot of errors. For example, when you first entered the app, you could type in credentials, but you could also just press the login button. The app in return would show the message “Welcome Jane Doe!”, but never take the user to the homepage. It was clear that the app needed a more security system for logging in. The register page would not open properly when the button was clicked originally often just crashing the app. Although the display of the activity\_register.xml file was designed similarly. The “remember me” checkbox button was also not functional and the hyperlink of forgot password was not connected. The user was never able to get to the home page of the app, it would often crash and go back to the main login page or the default android home screen. Therefore, the user never got to see any of the navigational pages.

If you look at the xml files within the old version of the app, it is clear that nothing had clickable functions nor was anything clearly connected. Permissions were also implemented within the notifications page which isn’t correct format wise. Permission requests often occur when an app is downloaded or a new feature is clicked on. Overall, the original version of the app had clear functionality issues and even had too many files in which many did not get used.

**Enhancements (Current Version)**

Enhancements were first added to the login features of the app. Seeing as there were a variety of security issues residing here. With research, I found out that firebase could be implemented with android studio in order to track and maintain user information. So now, when a user goes to log into the app, the credentials must be valid. If they are not valid an error message will appear and will not allow the user to move forward. The register page is also now accessible, the user can also implement credentials into it. Once all credential fields are filled in then the user will be able to log in at any time or log out. As for the forgot password hyperlink, this was converted into a button. The forgot password button also works through firebase. So once the request is sent with a valid email address that has been registered then a password request will be sent to the junk mail of the user. So, in terms of login features more database information was implemented and if/else statements were implemented in order to make the app more secure and increase the flow for the user.

Once the user logs into the app, it will not crash. With increased knowledge on how to use inflater and call upon the correct classes. We can see that the user can go from the login page to the home page without any issues as long as all credentials are valid. From here, an array was implemented to view events and scroll views were included so that the user can look through a variety of different events at one time. Building onto the algorithms and structures. The favorites page would work similarly, but is not running at this time because of issues with the inflater. Although, this would work similarly to the adapter for the events in which the favorited events would be placed into an array. They would be removed from the page based on the users’ decisions based on a button.

Lastly, the settings page is also reachable, in which shows some settings information. Although this still needs to be connected to the database to show the user’s actual name. Along with their profile picture.

**Software Engineering**

This component is mainly seen as the project expanded in terms of complexity. Originally the project was fairly simplistic and did not include login features with a database nor did the pages flow together. Although, we will talk more about the page flow in algorithms and structures. The app display on the homepage originally was just a generally layout in which the user could not interact with. Nor did any of the picture’s display or have information. Now, the user can scroll through the events listed in the home page. They are also able to click the favorites button in order to move these events into the favorites fragment. Originally, the dashboard was identical to the home page so these pages have gathered some complexity in terms of interaction with the user and the overall code.

**Databases**

The original version of the app did not include a database while enhancing this artifact I chose to implement one. Both in terms of the login features and how the event information was stored within the app. The login features store user credentials so that the user can always come back and log in with their information at any time. The user is capable of logging out of the application, registering, and changing their password. In order to achieve this, I implemented firebase into the application. Firebase stores the user’s credentials so that they may complete these activities. So once the user goes to the register page and enters their credentials then the app will log them in. From here they can utilize any of the features and if they choose to log out then the app will take them back to the log in page.

As for event information, this was more complex. This may not be working properly due to the login feature although I am not entirely sure. I had made a mock up version of the code to test it, but it has some confliction with the inflater. This code right now is unreachable purely because it will crash the application if the page attempts to connect to it. So, for now, it remains unattached. Although, this information is stored within SQLite. Originally, I had it stored with Firebase, but due to obstacles I switched which type of storage I was utilizing for the project.

**Algorithms and Structures**

The main enhancements here involve the page flow. The artifact originally did include the bottom navigation bar although it would crash periodically causing it to not flow properly. The login, register, and forgot password page all flow with each other within the updated version of this app. As well as the when the user logs in the login feature will now take them to the home page of the app in which they can now navigate through all pages listed on the bottom navigation bar. The only page that does not include clickable items is the settings page (Aside from the sign out button, which will take the user back to the sign in page). Another feature that helps to increase the structure of the app is the ability to add and remove events from the favorites page by interacting them on the home page. This is not fully completed but is partly implemented in the app due to some obstacles.