**Final Presentation III Overview**

**Grant Saylor, Kyle Smith, Anthony Tran, Jiayi Xu**

**6/7/2021**

**Topic:**

***Total Runtime: 15-25 minutes***

* **Subtopics:**

**Presenters:**

* ***Anthony***
* ***Grant***
* ***Kyle***
* ***Jiayi:***

**Transcript:**

***<Grant Saylor ->***

**1 min – 5 mins:**

***SLIDE 1 – TITLE SLIDE***

**Hi everyone, this is Grant Saylor speaking. We are Libraworks and our app is Virtual Library. In this presentation, myself, Kyle, Anthony and Jiayi will present demos to showcase the refinements that have been done since the end of Capstone 2. First though I’ll start with a refresher of what Virtual Library is.**

**<NEXT SLIDE>**

***SLIDE 2 – A REFRESHER ON VIRTUAL LIBRARY***

**As you may remember, Virtual Library is an Android application that is used to check out books from your neighborhood little libraries. The value add of Virtual Library is the ability to see libraries on a map and the collection of books contained within, all without having to leave your house. When you’ve decided on a library to visit all you have to do is utilize the barcode scanner to check out a title, this lets other users know that your book is not available.**

**The development team consists of myself, Grant Saylor, along with Kyle Smith, Anthony Tran and Jiayi Xu and is designed in Kotlin for Android devices. Each individual screen in our app, known as activities, has their user interface designed in XML.**

**<NEXT SLIDE>**

***SLIDE 3 – WHERE WE ARE IN DEVELOPMENT***

**Since you last saw a showcase of our app, at the end of capstone 2 we had implemented a rough version of most technologies that we wished to use. Since then, we’ve refined implementations by teaching ourselves more modern ways to implement functionality, such as infinite scroll instead of button presses to change pages. The app is in its release candidate form and we’re ready to publish to Google Play.**

**<NEXT SLIDE>**

***SLIDE 4 – TECHNOLGIES AND METHODS USED TO MANAGE***

**During the development of Virtual Library, we utilized a SCRUM-like development process where we met daily at 2pm to work on our project. It was important for us to simulate an in-person capstone experience, so to achieve this we created a server to discuss, share documents, video conference and more to replicate being at the same table in the Bellevue College lab rooms. This was extremely beneficial to us as it allowed our work to be supremely organized and always ready to find at the touch of a button.**

**Additionally, we each had our own branch on GitHub for each piece of the project which we would then merge into the main branch as a stable build. In conjunction with GitHub, we used Jira, a sprint management tool to divide the work into tasks so we were never too behind in our work.**

**<NEXT SLIDE>**

***SLIDE 5 – OVERVIEW OF NEW TECHNOLOGIES***

**In capstone 3 we have added various new technologies to Virtual Library such as infinitely scrolling pages, map pin clustering for visual cleanliness, animation to our app, which you can see on the right – implemented via API calls to OpenLibrary, cross activity connection for better ease of use, warning pages for loss of connection and fuzzy search, which you saw in our tech talk this quarter. A demo showcasing the major features of the app will follow.**

**For our first demo I’ll be handing it off to Anthony to showcase his primary work on virtual library, the searching screen.**

**<NEXT SLIDE>**

**<TAB INTO EMULATOR>**

***<Anthony Tran ->***

**10 mins – 15 mins:**

***Hi My name is Anthony and I’m going to be the one starting off our virtual library demo.***

***It’s been a while since we last presented our android app, so to refresh your memories, let’s start off with the book search activity in our app.***

***To begin, I want to let you guys know that we’ve used an open-sourced API called “Fuzzy Wuzzy” for the main logic behind this part of the application.***

**(taps search for books)**

***When searching for books in our application, we ask the user to type in the book title they're looking for or the author of that book.***

***Let’s type in Naruto, vol. 1 in the search bar***

**(Types in Naruto, vol. 1)**

***As you can see typing in Naruto, vol 1 will populate the activity with books that are like the user input. If you have not watched our tech talk, our search logic utilizes fuzzy search algorithms to query and search for books in our database based off of the user input. Going back to what I mentioned in the beginning, Fuzzy Wuzzy was implemented and used here. Also, the logic puts books that are more similar to the user input higher than those that are not as similar.***

***Now let’s type in Naruto 5***

**(Start typing naruto 5)**

***if just type in naruto 5 vs the previous naruto, (comma) vol. 1, the fifth volume of naruto would be placed higher in the view because its title is much closer to the user input versus all any of the other volumes.***

***It also can try distinguishing misspelled words and names using the fuzzy search algorithm to find a result. Let’s try typing in this:***

**(Starts typing joost az i em)**

***As you can see, misspelling just as i am with joost az i em can query and return the book fine.***

***Also, like i said before, users can search books by author as well.***

**(Starts typing in kishimoto)**

***And like with the book titles, the fuzzy search also applies for authors***

**(Starts typing sisly teson*)***

***Now that we know we can search for books in multiple ways, lets see what happens when we tap on a book.***

***First let’s type in “opal” to find the book “Artemis fowl and the opal deception”***

**(Starts typing opal)**

***Now let’s tap that book.***

**(Taps Artemis fowl opal deception)**

***A text box will populate the screen with at most three libraries that has this book in their inventory. The libraries shown are geo fenced so that only libraries that are about 5 kilometers away from your current location can be shown***

***If we click on “Kyle’s library”, the application will then leave the search activity and will be transferred to the maps activity, with Kyle’s library automatically pinpointed for the user to interact with.***

**(Taps Kyle’s library as I mention it)**

***Now I’m going to hand it off to Grant to talk more about the Maps activity.***

**<NEXT SLIDE>**

***<Grant Saylor ->***

***SLIDE 8 – Demo: Maps***

**Hi, it’s Grant again. I’d like to demo my primary focus on Virtual Library, which was the map screen. This activity utilizes the Google Maps SDK for Android to render the map technologies and is something I self-taught myself to create the implementation you see. As you saw, Anthony tapped on a library from his search screen and it zoomed into that specific location. Alternatively, this map screen is also accessible from the main landing page, however it will zoom into your GPS location. If I zoom in and out you can see that libraries will cluster themselves to make the map a bit more readable as more and more libraries get added. I’ll actually move myself to my current geolocation by tapping the button in the upper right.**

**<TAP GEOLOCATE BUTTON>**

**Now we’re in my neighborhood, Capitol Hill, Seattle. I’ve got a couple libraries around me such as <CLICK VIOLET> Violet and <CLICK BLM> Black Lives Matter. These names are derived from what the owner named them on the physical box.**

**Lets say that I wanted to check out a book from a library not near me, but still in my neighborhood. If I go there often I can add that to my favorite libraries and quickly get there with a tap**

**<TAP FAV LIB CENTRAL LIB> now I’ve zoomed over to this library a few blocks away, but I notice that there are only two books here and recently there hasn’t been much I’ve enjoyed, I think I’ll remove this library from my favorite list so I have room for another.**

**<LONG PRESS CENTRAL TO REMOVE>**

**<ZOOM BACK HOME> Well, that was a bust, but not all is lost, because there is much more you can do with the map screen. I think I want to create my own little library!**

**You can do this easily by naming your library up at the top <GEOLOCATE AND NAME IT HARVARD AVE LIBRARY>, now that I’ve given it a name I can reposition it exactly where the box is and confirm it <CONFIRM IT> now because this is my library I think it makes sense to add it to my favorites**

**<ADD TO FAVS, TAP ON IT>. As you can see, the map screen is a useful and powerful tool for navigating books in your area.**

**Next I’d like for you to all see Jiayi’s focus on Virtual Library, the library screen. Let’s zoom over to one of my favorites, Kyle’s Library to see a demo of that <GO TO KYLES>**

**<NEXT SLIDE>**

***<Jiayi Xu ->***

**10 mins – 15 mins:**

**Hi, this is Jiayi. I am going to introduce the library view screen. The functionality of this page is to show each library's information, such as the library's title, and book covers, etcetera. On top of the screen shows the name of the library that the user has now selected. And the user can see which books are in this library,**

**and these books are presented with covers and available copies, also the title of the book.**

**Therefore, it is easier to see if the library has the books they want. Some books do not have covers, so we will use blank covers instead of real book covers. (You can see some of the covers show with brown covers.)**

**If you want to take away the specified book, you need to go to the checkout screen to do checkout. Similarly, if you want to add some books to this library, also need to go to the checkout screen to do check-in. (Kyle will explain this part in more detail.)**

**The available book copies will update asynchronously when the user doing check-in or check-out.**

**You can see that there is a delete button in the lower right corner, and only logged-in users can delete their library.**

**To make this screen better, our group made this page scrollable instead of multiple views. To make this screen scrollable, we used the Grid View and it will create grids dynamically, and no need to worried about how many covers in each library. When we add a book or delete a book every time, it will dynamically add a grid or delete a grid at the same time.**

**To implement this page, the technologies of this page are reading data from Firebase and using Grid View to present data.**

**Now, I will hand it over to Kyle to introduce check-in and check-out. Thank you.**

**<NEXT SLIDE>**

***<Kyle Smith ->***

**5 mins – 2̴̨̮̇0̶̞̃ m̴̢͎̱͖̰̳̪̗̼̘̰̹̠̃͒̽͜͝i̵̦̞̼̲̘̪̾̍̈́̃̓͋̍̑̋́͛̌̿̚l̸͇͍̟͍̜͇̺̩̺̖͔̥͙͈̊̀͐̅̓̋͋̽͛̄̄͗̚̕ͅl̴̲̼̙̮̻̪̮̜̲͇͎͑̓̎̅̃́͐i̷̲͙̺͖̼̤͇̥̐̄͌̓̔͊́̔̕̚͝͠ͅo̶̧̡̠̲̘͚͙̲̻̯̺̲͙͌́̓̓̚͜n̷̛̮͙̦͊̆̆̍͑̊̿͋̏̔̌ ̵̢̬́͑̈̒̏̆̽̈̽̓̓͐̇͋͆̅̃̐͂̒̀̋̔̈̈́͛̓̉̇̐͗̕̚̚̕͝ý̴̢̧̡̨̙̣͓̘͚̝̟͕̜͓̻̠̹̟͍͕̻̣͉̝̱̟͈̪̹̻̗̱̞̺͕̰̥͓̤̘͊̓́̒̋̍̇̎̔͋̃̑̂̃͐̆̄̾̈́̌̊̈́̚̚͜͜͝͝͠ͅę̶̡̨̛̛̗͕̙̤̱̮̟̖̞̼̱̖̖̮͚̰̱͕̮͔͚̣͈͖̼͕̰͇̪̪͍̺̜͕͚̩̬̂̎̑̓̉̎̒̎̀͒̈̇̽͆̾͗̎̌͌̈́̏̾̂̆̔͒́̒͛̂̓̇̀̅̕͜͜͜͠͝ͅą̴̨̢̧̧̜̬̝̬̳̜̗̲̲̰̹̲̰͇̻̜̩̻̼̳̣͍̺͎̱̬̙̺̪̞͎̖͎̖̯͒̄̇̃̀̋̋̾̓̈́̅̔͗̏̒͝͝ͅṟ̸̟̬͈̅̽̆s̶̡̧̨̡̛̫͓͖͚͉͍̲͕̮̻̝͕̯̺̑̓͑̿͋́͌̽͛̈́̓̿͐̅͊͑̀̾͗̽̀͆̏͒̒͗̂̃̄͊̈́̃̊͗̌̃͋̌̂͐̂̚̚͝͝͝:**

**Hi everyone, this is Kyle and I’m going to be covering the final parts of our app, the barcode scanner and checkout log.  This portion of our app implements Google Vision as a way to interface with the device’s camera for scanning a barcode, and also communicates with OpenLibrary, a free online book website dedicated to cataloging every book ever published. This page is the checkout screen, and is where the user will use their device’s camera to scan a book barcode and then press a button for the app to correctly update the database for what the user is doing.  When the user arrives at this page, the activity determines where the user came from previously, and that tells the activity whether this transaction is happening in a user’s collection, or within a library in our database.  This information will play a role when the user taps the checkout button at the bottom.**

**\*Gesture at the video feed within the frame\***

**The black box within the frame is actually a video output for the device’s camera.  We use Google Vision to detect when a barcode is in view of the camera, and then if the scanned barcode was for a book, we now can get the ISBN.  Once that ISBN is obtained, we send out a call to our book API, Open Library, and grab pertinent information for our own DB, such as the title, author, publication date, and cover image.  We populate the title of the book on screen to indicate to the user that the barcode has been scanned.**

**<GO TO VIDEO>**

**<Freestyle> Now the user can tap on the button to indicate whether they are checking this book into a library, or checking it out.  Once they have that set, they can tap the button at the bottom to complete the transaction.**

**<AFTER VIDEO>**

**At this point, all the book information is being sent to our database, and as that completes, we send a toast to the User and display the cover image for the book to indicate that the process completed.  For our DB, we store information in a few different places to help keep things organized.  We keep the complete records for each book in one branch of our DB, while just using the ISBN of the book as a key in other branches, such as the one for our libraries.  For instance, the branch of our DB holding information about each library doesn’t need to know the publication date for the books, because that information is already stored elsewhere.  We can simply keep a list of all the ISBNs registered at a particular little library, and when we need the information for those books, we can reference the other branch in our DB.  If the user is checking in a brand new book, we create a new entry that holds all the book information, and then we update some values in the library branch, to indicate that it has a new book in stock, and we set it’s number of available copies to 1.  We have a series of parameters for each transaction that this activity is able to handle, things like if the book is new to our DB or not, if the book is being checked in or checked out, and if the transaction is taking place in the user’s collection or the library’s circulation.  Now that I’ve shown you the check-in process for a book, I want to show you the check-out log.**

**\*set the button to “Check Out”\***

**When a user checks out a book from a library, we add that book, as well as the date and the library name, to a list that is maintained for that user.**

**\*Tap the checkout button at bottom, then head to the user profile screen, and click on checkout log\***

**So if we now check this book out, we can then go back to the User Profile activity within our app, enter the Checkout log by tapping the button at the center of the screen, and now we are presented with a scrollable UI that tells us the “what, when and where” for all the books we have checked out.  Once we return one of these books and scan the barcode at the library that we borrowed it from, it will be wiped from the checkout log.**

**<NEXT SLIDE >**

**So, as with any project, we had our fair share of challenges over the 3 quarters we spent on this mobile app. Firstly, a major component of our app we wanted to include was the ability for users to directly share books with one another. However, this presented us with a multitude of challenges, with solutions generally making our app feel more like a social media platform than a companion app for Little Free Libraries. In the end we had to drop this functionality as it would have cost us too much development time and would have pushed our project astray from its roots. Another major hurdle for the team was the entire process of working on a mobile app, itself. None of us had any prior teaching, training, or know-how for working with Android Development, and so even doing the very basics like finding which files we needed to write code into, or just understanding how all the many pieces fit together to make the whole required a bit of homework at the beginning. Even as we put some finishing touches on our project for this capstone, we’re still learning about different ways of accomplishing tasks in this environment. Lastly, some of the tools we used to bring this project to life were a little difficult to work with. Specifically, OpenLibrary, as we found, is not super consistent with the way that they clump information together for each book. When we parse this information, we tried using RegEx, and web scraping before settling on utilizing JSON formatting, but we still ran into issues relating to empty fields, different data formatting, and more. This took me, personally, quite some time to figure out to ensure our app could handle the variety of books users will eventually scan in.**

**Despite all the challenges we faced throughout capstone, I have to say I’m extremely proud of what this group has accomplished. I never would have expected our creation to be what it is today, and I hope you all have enjoyed following our journey this past year.**

**<NEXT SLIDE>**

**These are all the sources for the different tools we used for our project, most of which we highlighted during our different parts.**

**<NEXT SLIDE>**

**And these are all the sources and licensing information for all the graphics and art used in our application.**

**<NEXT SLIDE/FINISH PRESENTATION>**

**Thank you for watching our presentation, we’re ready to answer any questions you have.**