CS443: Mobile App Development in Android
21 December 2021
Anna Boccuzzi
Matheus Pereira

Final Project Report

Project Statement

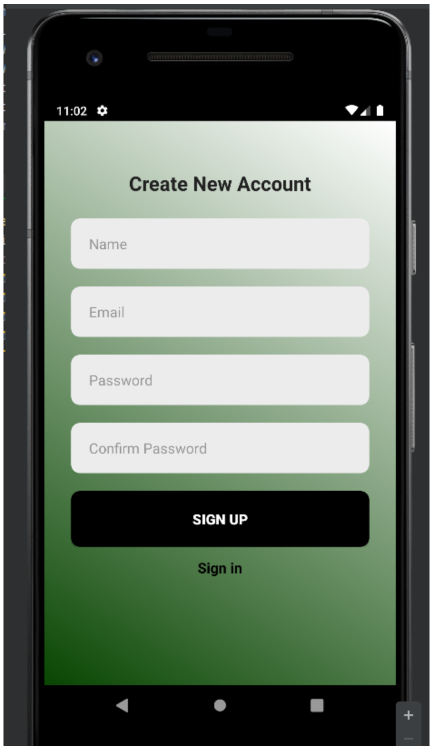
A finance management app. The app has multiple aspects of financial analysis included such as a transactions dashboard, a saving goal chart, where the user can input their goal dollar amount to save, how much they have already saved, and their intended monthly contribution. The transactions dashboard is useful to keep track of your income and expense transactions. It also keeps the total amount for your income and expenses, respectively. The chart is colorful, interactive, animated, and customized. The app includes other features including a calendar tab that displays the current calendar month and day, scrollable vertically in contrast to the app's horizontal tab scroll feature. The user may click on certain days to add them to a list of days saved below the calendar as due dates for bills. There is a coupon tab which has four icons displayed by imageView elements, all of which are clickable and link the user to outside discount websites that encourage financial savviness. The app is a desirable topic because it is a useful app, possibly an educational app, and offers interaction with many different java and xml features the class did not yet reach. The sign in aspect of the app also introduced experience to database creation and management. The sign in feature additionally provided practice and challenge in connecting multiple activities. The app has no special requirements and can be used on any mobile phone.

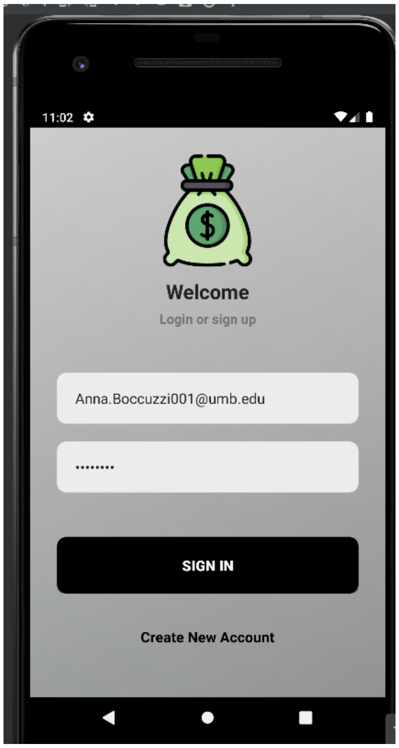
Application Design

The app has three main activities including the sign up activity which allows the user to make a profile that is saved to the Firebase database, the sign in activity that validates the login and launches the third activity, and the third and main finance management activity. The finance management activity is an activity formatted in tab layout drawing on the sample code EffectiveNavigation provided to the class as a sample code. From there the sample code was edited to change the icon to fit the theme, the title to fit the app idea, and the tabs made with customized names and a greater number of tabs. The sign in sign up page features a unique animated background that sophisticates the appearance for users, and this animated background is also featured in the later tabs. The tabs are scrollable on the horizontal axis from left to right, the user may drag on the titles on the tabs, or may swipe on the main fragment displayed from each tab title. The tab titled "transactions" keeps track of all transactions the user puts in. Once the user puts in an income or expense, the app checks for changes in the database and updates the total income and expenses. There is also a recyclerView to list the recent transactions the user puts in. The tab titles saving goals has a blank pie/donut chart to start with, three edit texts for the user to fill in their goal amount, amount already saved, and monthly contribution. After filling in these amounts the user clicks the 'OK' button to animate and fill the pie chart with their data, which also displays in a textView at the bottom how many months it will take to achieve their goal. The next tab entitled billing cycles features a calendar view that is scrollable vertically and displays a wide range of months with the weeks of the year on the far left and a highlighted selected day display. When the user clicks a day it is added to their list of bill due dates directly below the calendar. The coupons tab has four imageViews with icons that represent some financial element, all of which are functionally clickable and link the user directly to web pages that open in the browser. All of the links are popular coupon websites that may help the user find deals and discounts. The app is well suited for both phone or tablet, on a phone the tabs need to be scrolled whereas on a tablet the tabs would likely all be visible. The savings goal app responds to a button click to take the user input and create the chart using an onClick listener. The calendar tab updates the list of due dates with an onSelectedDayChangeListener. The coupons tab responds to onClick actions with an onClick method referenced in the xml file and a tag holding the websites in the xml, the method defined in the main activity java to launch the website.

Application Implementation and Evaluation

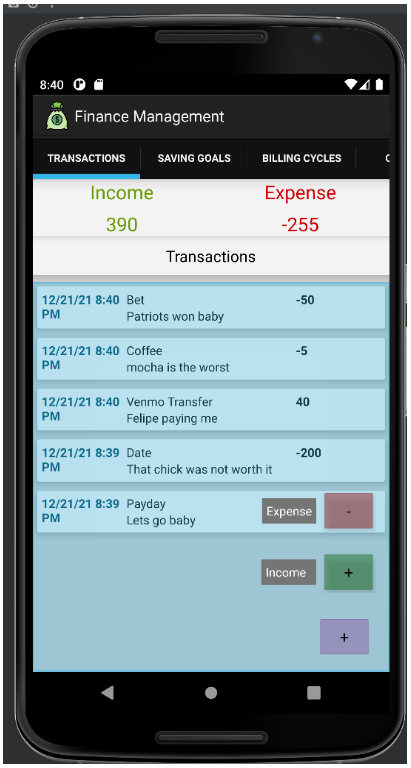
The sign in and sign up page depend on the user's information being stored in the database to validate if the user has matched a created login when clicking on sign-in. Both java files utilize on create methods and validation methods, as well as other ones such as loading, showToast, and setListeners. The sign up xml has four editTexts for name, email, password, and confirm password, a button to submit the info, a progress bar, and two textViews one on top and one on the bottom. The sign in xml features an imageView with the app's icon, two text views instructing the user, two editTexts for email and password, a button to sign in, progress bar, and text view that links to the sign up if the user does not have a profile yet made. The main xml is only a pager as was set in the professor's sample code, while the mainActivity java code has an onCreate to set up the pager and action bar, an onTabSelected, onTabUnselected, and onTabReselected, the AppSectionsPagerAdapter class with a switch that launches the specific fragments to the tabs respectively. There is also a getCount for the number of tabs, getPageTitle to set the tab titles, and an openBrowser method that is used by the coupons xml. The transaction fragment has a linear layout having the income and expense viewTexts at the top. They change whenever the user opens up the floating button to add an income or an expense. The floating button has an animation design and has two other buttons pop up. Once either the income or expense is added, the data goes to firebase. The income and expense total amounts at the top have a listener for any change in the database to be updated. Also, whenever income or expense is put in, the recycleView gets updated and is seen on the dashboard. The savings goal xml has a constraintLayout with a pieChart on top, three editTexts for user input, a textView that displays a calculation based on the user input, and a button for the user to notify the app they have finished entering info. The java for the savingGoals has an onCreateView inside the savingGoals class that extends fragment class. Additionally there is an onViewCreated section that has an onClickListener to respond to the button click and parse the user input. The pieChart aspect uses a setUpPieChart method, makeEmptyChart method for the app's initialization, and a loadPieChartData method. The billingCycles xml has a constraintLayout with a calendarView, and another nested constraintLayout with a textView. The java for billingCycles is made of a class called BillingCycles that extends fragment, an onCreateView, onViewCreated, and a setOnDateChangeListener. Inside this method more textViews are created and populate the app when dates are clicked. The coupon tab xml has a constraintLayout with textView title and four imageView icons. The imageViews all have tags with websites and an onClick method that uses the tag to launch the websites on the phone’s browser. The coupon tab’s java has a simple onCreateView. The app was tested a plethora of times in between virtually every change to confirm the changes would allow the app to still build and run without crashing. The debugger was used to identify problems such as build problems, calculation errors, performance issues, etc. The app works correctly reliably, though the calendarView’s matching list that is created when dates on the calendar are selected occasionally is sensitive to scrolling the calendar and populated the list with the same day multiple times, though with a careful hand this is avoidable. The savings goal section was tested to correct the math, truncate the float value, display the text in a readable font size, etc. The calendar xml was challenging to format and was not very customizable with the majority of defined functions on the android developers webpage being deprecated. All of the issues were resolved though there is always room to improve.





Anna.Boccuzzi001@umb.edu

••••••••



Income	Expense
390	-255

Transactions

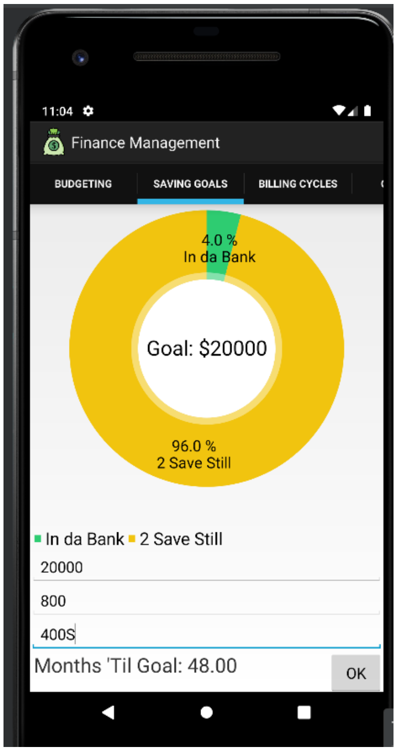
12/21/21 8:40 PM Bet -50
Patriots won baby

12/21/21 8:40 PM Coffee -5
mocha is the worst

12/21/21 8:40 PM Venmo Transfer 40
Felipe paying me

12/21/21 8:39 PM Date -200
That chick was not worth it

12/21/21 8:39 PM Payday
Lets go baby

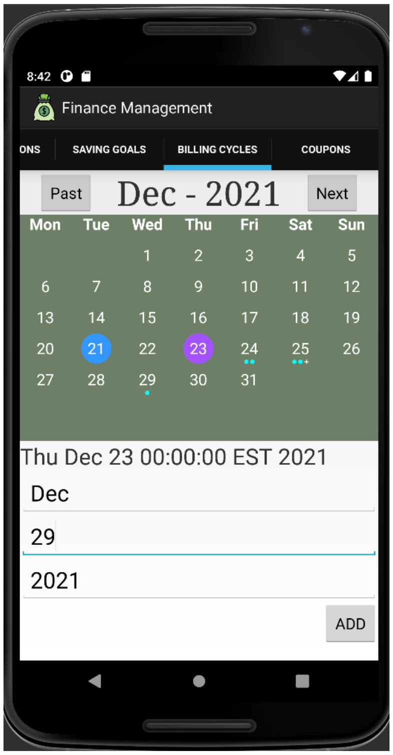


20000

800

400S

Months 'Til Goal: 48.00



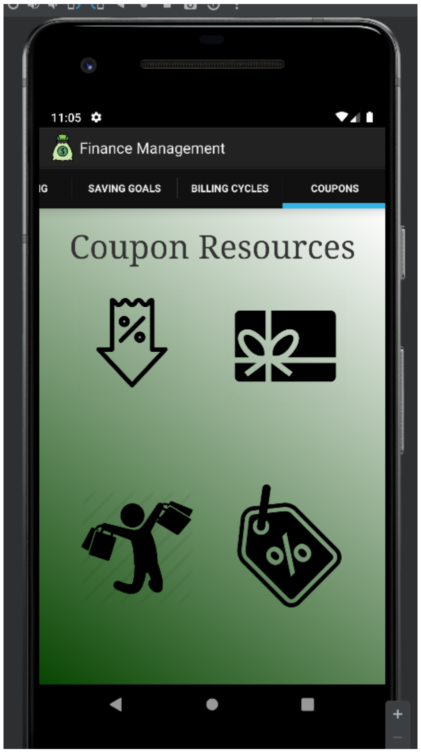
Mon	Tue	Wed	Thu	Fri	Sat	Sun
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

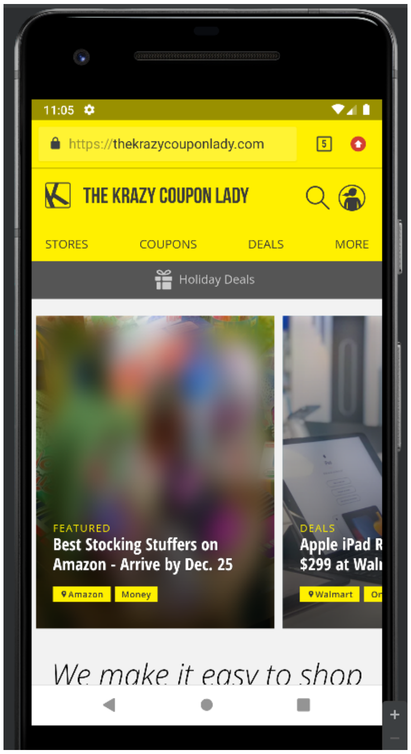
Thu Dec 23 00:00:00 EST 2021

Dec

29

2021





Best Stocking Stuffers on Amazon - Arrive by Dec. 25

Apple iPad R $299 at Wal

We make it easy to shop

References

Effective Navigation (Professor's Sample Code)

How to Create a Pie Chart in an Android App with MPAndroidChart

Pie Chart not displaying in a fragment(Android)

Github PhilJay MPAndroidChart

How to add a calendar in a fragment on Android Studio?

Android Beginner Tutorial #26 -CalendarView

Android Developer Reference Calendar View

Can't change Android CalendarView to another month (Samsung S Duos - Android 4.2.2)

Creating a text view on every button click

How to create a list of Textviews

Android - Set text to TextView

ImageView displaying in layout but not on actual device

Add web link to ImageView?

Round to 2 decimals and pass through TextView

Expense Manager App - Part 1 Create Project and Login Layout Design

Animated Background | Android Studio

Android Chat App Development | Tutorial #4 | Sign In Using Firestore Database & Shared Preferences

SundeepK/CompactCalendarView

Experiences and Thoughts

The project was a great opportunity to pursue android concepts the class did not reach yet, for example the database, multiple activities, pie chart, calendar view, clickable images, and more which were not a part of the previous homework assignments. The transactions took a great deal of time. Learning how to implement all types of firebase features and how to make the fragment work properly when there is a lot going on. Specifically, how to make the recycleView work with the FirebaseRecycleAdapter and put it on screen. The calendar view was a learning point where after implementing the calendar we learned that the android widget does not allow for adding events, which led us to problem solve a way to display events going around this missing element, the solution being a simple list of the dates that would have been added as events. Connecting the activities was also a great challenge that required much time and effort into problem solving when the errors do not accurately describe the problems in the code. The saving goals tab was a great colorful aspect and taught us how to work with graphs which were unique and unseen yet in android studio in our learning journey. The connection of the user input populating the graph with an animation was a major success in our implementation, as well as the issue of truncating the month value to two decimal places which was more difficult than expected as multiple strategies proved ineffective in our development experience. The coupon tab was an effective learning experience on formatting images and how the design display for xml is sometimes misleading, in which case one must be confident in their code to ensure a proper display and result. The calendar view could have been improved with a custom calendar class, though this was not our focus to invest time into.

Submission

Create a github project and upload all your materials there, and submit the github link on blackboard. The submission includes the source codes and the project report file. Please also make a README file that includes the instructions of how to compile and run your app. Test your app well before submission.