

FinanceSmart

by

Shiv Joshi and Devansh Patel

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright by 2020 Devansh Patel | Shiv Joshi

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Shiv Joshi _____ 04/12/2020
Date

Devansh Patel _____ 04/12/2020
Date

Abdou Fall _____ 04/12/2019
Advisor Date

University of Cincinnati

College of Education, Criminal Justice, and Human Services

April 14, 2020



FINANCESMART

Financial and Credit Application



TABLE OF CONTENTS

- Abstract.....3
- Introduction.....4
 - ◆ Problem Statement.....4
 - ◆ Solution.....5
- Project Management.....6
 - ◆ Project Objectives/Goals.....6
 - ◆ Project Scope.....6
 - ◆ Budget.....7
- Project Timeline.....8
 - ◆ Work Breakdown Structure.....8
 - ◆ Gantt Chart.....12
 - ◆ Use Case Diagram.....14
- Technical Discussion.....15
 - ◆ Network.....15
 - ◆ Development Application.....15
 - ◆ Database.....15
 - ◆ Architecture & Technology Used15
 - ◆ User Interface.....16
- Test Plan.....17
 - ◆ Importance of Test Plan.....17
 - ◆ Overview, Product Details, and Users.....17
 - ◆ Scope of Testing.....18
 - ◆ Testing and Reporting.....18
 - ◆ Assumptions – pass/fail.....18
 - ◆ Test Cases.....18
 - ◆ Future Recommendations/Recommendations for Improvement.....20
- Conclusion.....21
- Works Cited.....22
- Appendix.....23
- List of Illustrations, Figures, and Tables
 - ◆ Table 1 – FinanceSmart Estimated Cost.....7
 - ◆ Table 2 – FinanceSmart Work Breakdown Structure.....8
 - ◆ Chart 1 – FinanceSmart Gantt Chart.....13
 - ◆ Figure 1 – FinanceSmart Use Case Diagram.....14
 - ◆ Figure 2 – FinanceSmart User Interface.....15
 - ◆ Figure 3 – Architecture and Technology Diagram.....16
 - ◆ Figure 4 – FinanceSmart Users.....17

Abstract

FinanceSmart is a Xamarin based mobile application targeted at middle to late adolescents in order to teach them how to be prepared for financial success with Credit Card debt and financial knowledge.

Credit Card debt and lack of financial knowledge has a negative effect on young adults for adulthood, most adults do not have enough capital saved up to cover them in any financial emergency or are so deep into debt that they are just trying to pay it back but don't know how to effectively go about it. Also, currently available applications don't have customizable features that fits user needs.

FinanceSmart will be a cross - platform application with an easy to use intuitive user interface to entice adolescents to take responsibility for their finances and financial literacy. Customizable budgeting features will keep users up to date on financial aspects.

Introduction

Problem Statement:

In 2013, in the Federal Reserve Survey on “monitor the financial and economic status of American consumers," When asked what if they have an emergency for \$400 nearly 47 percent said that either they would cover it by borrowing or selling something, otherwise they would not be able to make it. A 2013 Gallup Poll found out that only one in three Americans had a proper written budget. A Bankrate survey in 2014 found that only 38 percent of Americans are capable of covering \$1000 emergency room visit or \$500 car repair cost with the money they’ve saved. Similarly, a report by the Pew Charitable Trust found 71 percent households were concerned about covering only everyday expenses. Similar study by Annamaria Lusardi (George Washington University), Peter Tufano (Oxford), and Daniel Schneider (Princeton) found more than 25 percent could not make \$2000 in 30 days for an unanticipated expense and 19 percent could only if they pawned possessions or borrow payday loans. All above studies and reports conclude that about 50 percent of adult Americans are living very close to financial-edge and they need some ideas or training on the use of credit cards to level-up them from such conditions.

Because of Credit Card debt and lack of financial knowledge of young adults to be prepared for adulthood, most adults do not have enough capital saved up to cover them in any financial emergency or are so deep into debt that they are just trying to pay it back but don’t know how to effectively go about it.

Based on our research to this present day there are other similar applications such as Albert and Mint. Although other applications like Mint offer free credit check (soft pulls on your credit), they do not offer the customizability and features that FinanceSmart aims to offer, our focus is on adolescents who are just joining the workforce and/or who want the ability to create their budgets on a weekly/bi-weekly basis.

Solution:

FinanaceSmart is a Xamarin based mobile application targeted at middle to late adolescents in order to teach them how to prepare themselves for financial success. Our multi - faceted application will not only teach the user how to budget but it will also teach them how to use/build credit and track their spending.

FinanceSmart has a targeted demographic of middle to late adolescents. This includes college students, high school students, young adults (mid to late twenties), to help them navigate the financial challenges of adulthood. FinanceSmart will be a cross - platform application with an easy to use intuitive user interface to entice adolescents to take responsibility for their financial literacy. FinanceSmart is strong in its unique ability to be customized at the user's discretion to full their needs when they need.

Project Management

Project Objectives/Goals:

Develop a cross – platform application that will provide middle to late adolescents with the ability to learn, track, and integrate their finances all through one application.

FinanceSmart will suggest feedback on their spending habits.

FinanceSmart is a Xamarin based mobile application targeted at middle to late adolescents in order to teach them how to prepare themselves for financial success. Our multi-faceted application will not only teach the user how to budget but it will also teach them how to use credit and track their spending.

The features of FinanceSmart will include:

- Track spending habits which allows the user to visually see where their money is going and will provide feedback on their spending.
- In app budget planner to help the user to create a budget based on their needs and state in

In conjunction, all these features will provide a comprehensive experience for adolescents to better control their finances and increase financial literacy for youth.

Project Scope:

Our team is developing a mobile application that will help middle to young adolescents understand their finances for adulthood. FinanceSmart will achieve the project scope through these following milestones:

- Easy intuitive UI to entice adolescents to want to take control of their finances.
- Front end written in XAML
- Back end written in C#
- We will be using Xamarin Forms for code reuse.
- FinanceSmart will comply with all PCI Data Security Standards
- User access will be secured with the use of modern authentication tools

Budget:

Table 1 – FinanceSmart Estimated Cost, provides the estimated cost of building the FinanceSmart application including all development and equipment costs.

FinanceSmart Estimated Budget			
Particulars	Units/Hours	Cost per Unit/Hour	Total Cost
Manager Wages	150	\$25	\$3750
Developer Wages	300	\$20	\$6000
Support/Maintenance	1	\$2000	\$2000
Hardware/Computer	2	\$650	\$1300
Software - Xamarin	N/A	\$0	\$0
Total Project Cost			\$13050

Table 1 – FinanceSmart Estimated Cost

Project Timeline

FinanceSmart Work Breakdown Structure:

The FinanceSmart deliverables and Work Breakdown Structure are represented by the details contained in **Table 2 – FinanceSmart WBS**. These items represent the work needed to be completed for both the Fall and Spring Semesters.

Task Name	Duration	Start Date	End Date
1.0 Project Management and Deliverables	252	8/29/19	4/15/20
1.1 Team Building	2	8/30/19	9/1/19
1.2 Idea and Brainstorming	7	9/1/19	9/8/19
1.3 Fall Semester Assignment 0: Team Members & Project Name	3	9/8/19	9/11/19
1.3.1 Project Name	1	9/11/19	9/12/19
1.3.2 Project Logo and Branding	5	9/12/19	9/17/19
1.4 Fall Semester Assignment 1: Team Contract	8	9/17/19	9/25/19
1.4.1 Project Approval	2	9/25/19	9/27/19
1.4.2 Gantt Chart	4	9/27/19	10/1/19
1.4.3 Work Breakdown Structure	8	10/1/19	10/9/19
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	5	10/9/19	10/14/19
1.6 Fall Semester Assignment 3: Team Contract Resubmission	3	10/14/19	10/17/19
1.6.1 3 – Minute Elevator Speech	2	10/17/19	10/19/19
1.6.2 Elevator Speech Practice	3	10/19/19	10/22/19

1.7 Fall Semester Assignment 4: User Profile	5	10/22/19	10/27/19
1.8 Fall Semester Assignment 5: Use Case Diagram	12	10/27/19	11/8/19
1.9 Fall Semester Assignment 6: Draft Report	6	11/8/19	11/14/19
1.10 Fall Semester Assignment 7: Final Fall Semester Report	10	11/14/19	11/24/19
1.11 Fall Oral Presentations	6	10/21/19	12/2/19
1.12 Spring Semester Assignment 1: Testing Plan / Report	0	1/13/19	1/14/20
1.13 Spring Semester Assignment 2: Abstract	0	1/14/20	1/14/20
1.14 Spring Semester Assignment 3: Draft Tech Expo Poster	0	1/14/20	1/14/20
1.15 Spring Semester Assignment 4: Finalize Expo Poster	0	1/14/20	1/14/20
1.16 Spring Semester Oral Presentation	1	1/14/20	1/15/20
1.16.1 Oral Presentation Practice	2	1/15/20	1/17/20
1.17 IT Expo	1	1/17/20	1/18/20
1.17.1 IT Expo Exhibit and Preparation	1	1/18/20	1/19/20
1.18 Spring Semester Assignment 5: Final Project Report	1	1/19/20	1/20/20
1.19 Spring Semester Assignment 6: Safe Assign Final Project Report	1	1/20/20	1/21/20
1.20 Spring Semester Assignment 7: Finalize Library Copy	2	1/21/20	1/23/20
2.0 Research	2	1/23/20	1/25/20
2.1 Software Requirements	3	1/25/20	1/28/20
2.1.1 Understanding the chosen software tools	0	1/28/20	1/28/20

2.1.2 Finalize Back End Development Language(s)	0	1/28/20	1/28/20
2.1.3 Finalize Front End Development Language(s)	0	1/28/20	1/28/20
2.2 Network Requirements	1	1/28/20	1/29/20
2.3 Security Requirements	5	1/29/20	2/3/20
2.3.1 Research PCI Data Security Standard Compliance	0	2/3/20	2/3/20
2.3.2 Research Login and Credential Storage	0	2/3/20	2/3/20
2.3.3 Research Legal Disclaimers	0	2/3/20	2/3/20
2.4 Miscellaneous Research	2	2/3/20	2/5/20
2.4.1 Budget Analysis	0	2/5/20	2/5/20
2.4.2 Financial Literacy of adolescents	0	2/5/20	2/5/20
3.0 System Design	7	2/5/20	2/12/20
3.1 Create Systems Design	0	2/12/20	2/12/20
3.1.1 Create Network Diagrams	0	2/12/20	2/12/20
3.1.2 Create Database Diagrams	0	2/12/20	2/12/20
3.1.3 Create Wireframes	0	2/12/20	2/12/20
3.1.4 Create Prototype Mockup	0	2/12/20	2/12/20
3.2 Create Legal Documentation	0	2/12/20	2/12/20
3.2.1 Legal Disclaimers and Privacy Policy (Draft Document)	0	2/12/20	2/12/20
3.2.2 PCI Data Security Standard Information and Consent Form	0	2/12/20	2/12/20

4.0 Environment Set - Up	2	2/12/20	2/14/20
4.1 Import Libraries for Development	0	2/14/20	2/14/20
4.1.2 Get familiar with the Development Environment	0	2/14/20	2/14/20
4.2 Setup a Team GitHub Repository	1	2/14/20	2/15/20
4.3 Download Xamarin for Visual Studio	1	2/15/20	2/16/20
5.0 Development (Back End and Front End)	15	2/16/20	3/2/20
5.1 Create the Main Menu and Landing Page	0	3/2/20	3/2/20
5.2 Create Login and Set Up User Authentication	0	3/2/20	3/2/20
5.2.1 Create User Registration and Confirmation	0	3/2/20	3/2/20
5.2.2 Create Forgot Password	0	3/2/20	3/2/20
5.3 Design and Develop Features	20	3/2/20	3/22/20
5.3.1 Design and Develop Daily Reports and Update via Google Docs	0	3/22/20	3/22/20
5.3.2 Design and Develop CC Card and Debit Card Integration	0	3/22/20	3/22/20
5.3.3 Design and Develop Financial RSS Feed	0	3/22/20	3/22/20
5.3.4 Design and Develop Budget Functionality for the User	0	3/22/20	3/22/20
6.0 Testing	8	3/30/20	3/30/20
6.1 Create Mock Users	0	3/30/20	3/30/20
6.1.1 Create Desktop Users	0	3/30/20	3/30/20
6.1.2 Create Mobile Users	0	3/30/20	3/30/20

6.2 Regression Testing	4	3/30/20	4/3/20
6.2.1 Regression Test for Users	0	4/3/20	4/3/20
6.2.2 Regression Test for Budget Feature	0	4/3/20	4/3/20
6.2.3 Regression Test for CC card and Debit Card Feature	0	4/3/20	4/3/20
6.2.4 Regression Test for RSS Feed	0	4/3/20	4/3/20
6.3 Security Test	5	4/3/20	4/8/20
6.3.1 Firewall Test	0	4/8/20	4/8/20
6.3.2 Penetration Test	0	4/8/20	4/8/20
6.3.3 SQL Injection Test	0	4/8/20	4/8/20
6.4 User Acceptance Test	5	4/8/20	4/13/20
6.5 Final Certification Test	3	4/13/20	4/16/20

Table 2 – FinanceSmart WBS

Gantt Chart:

The FinanceSmart timeline is depicted in **Chart 1- FinanceSmart Gantt Chart**, and represents the work completed in both the Spring and Fall Semesters.

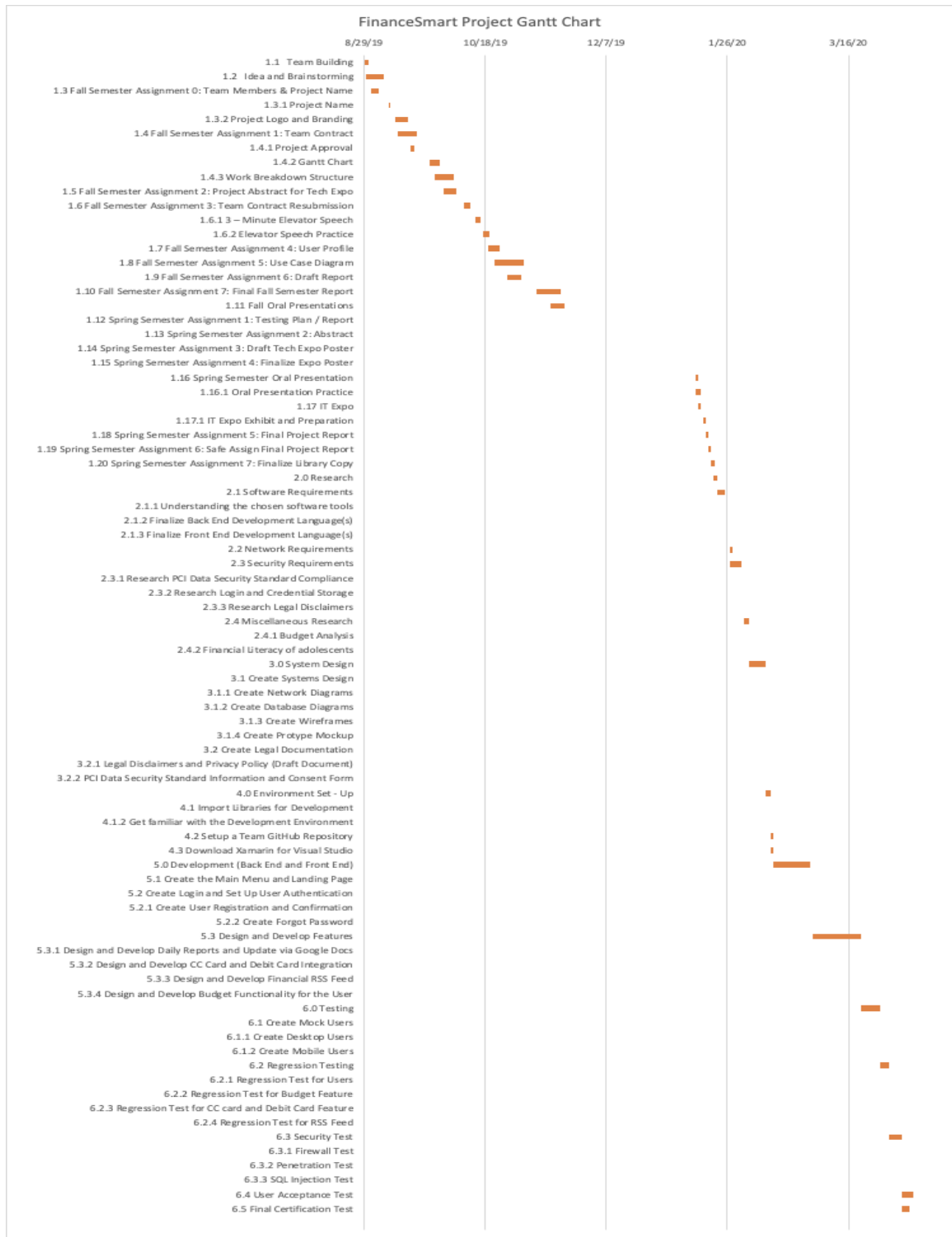


Chart 1 – FinanceSmart Gantt Chart

Use Case Diagram:

The Use Case diagram below, **Figure 1 – FinanceSmart Use Case Diagram**, depicts the three primary use cases for FinanceSmart: Teenagers, College Students and Credit Card Users.

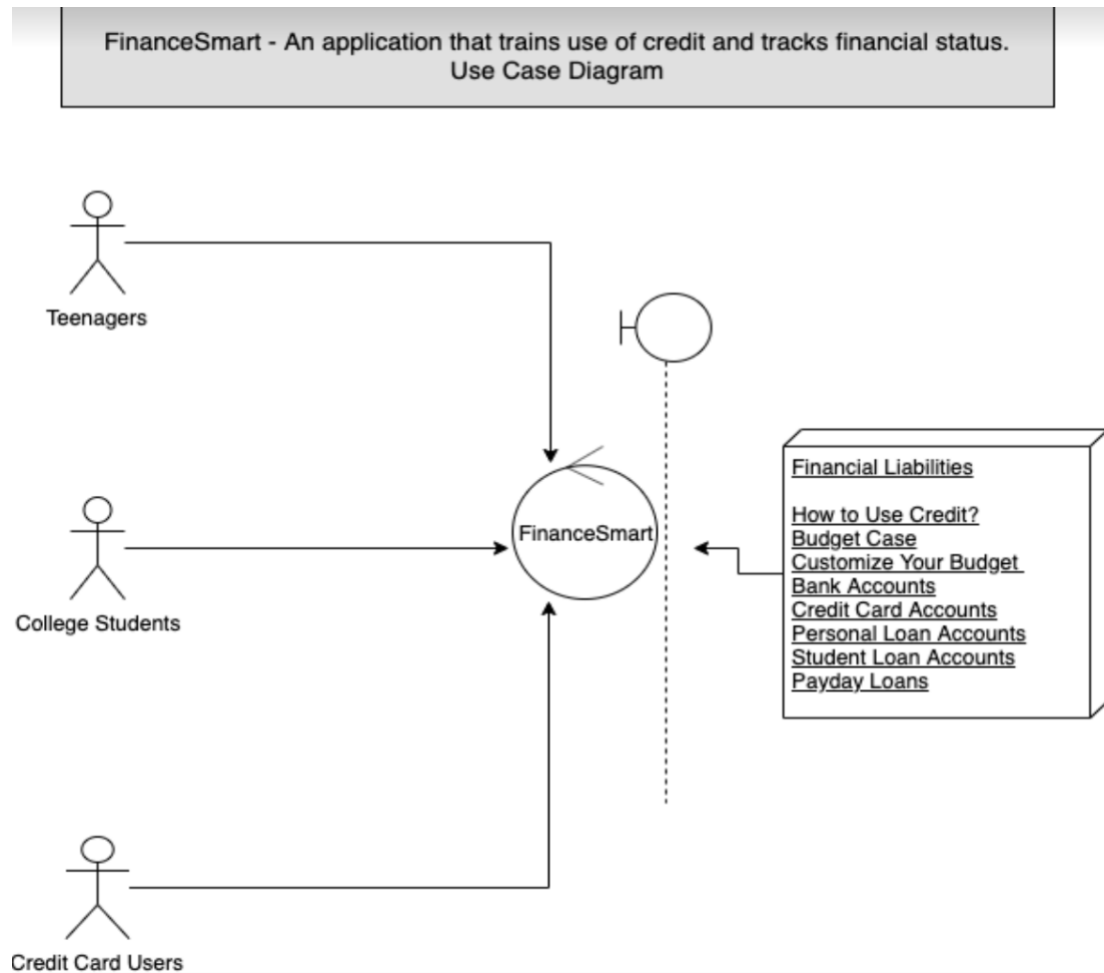


Figure 1 – FinanceSmart Use Case Diagram

Technical Discussion

Network:

We will design this FinanceSmart application to run in the network that has at least one server which will connect the application to the database server.

Development Application:

We will use ASP.Net to develop the application within Xamarin development environment which will be used in mobiles for ease of use to the end users. XML will be used to design and handle screens. Mobile front-end part of the FinanceSmart application will be handled by Xamarin forms that will give flexibility to be cross platform application within Android and iOS stores.

Database:

SQL Server will be used to run the application for database support where user inputs will be stored and retrieved as needed, depending on the button click events in the application. Back end will be driven with relational database architecture. For user authentication and security purpose Google service – Google Sign In will be used which will save time to develop sign in logic for developers. We expect user to have at least one active email address to register and be notified of activities.

User Interface:

Figure 2 – FinanceSmart User Interface, depicts the “New Item” and “Browse Item” screens that FinanceSmart users would utilize to add new items and search for existing items in their budget.

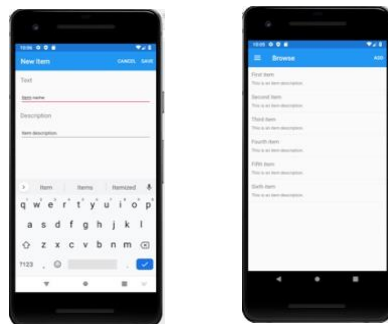


Figure 2 – FinanceSmart User Interface

Architecture & Technology Diagram:

Figure 3 – Architecture and Technology Diagram, depicts the application will be using Xamarin as development environment on .NET framework with C#, and XAML as Xamarin forms handler. Users and Admins can login with username and password then authentication will be handled with Google authentication. Data saved by users will be saved to SQL Server which is determined to be database server for the application.

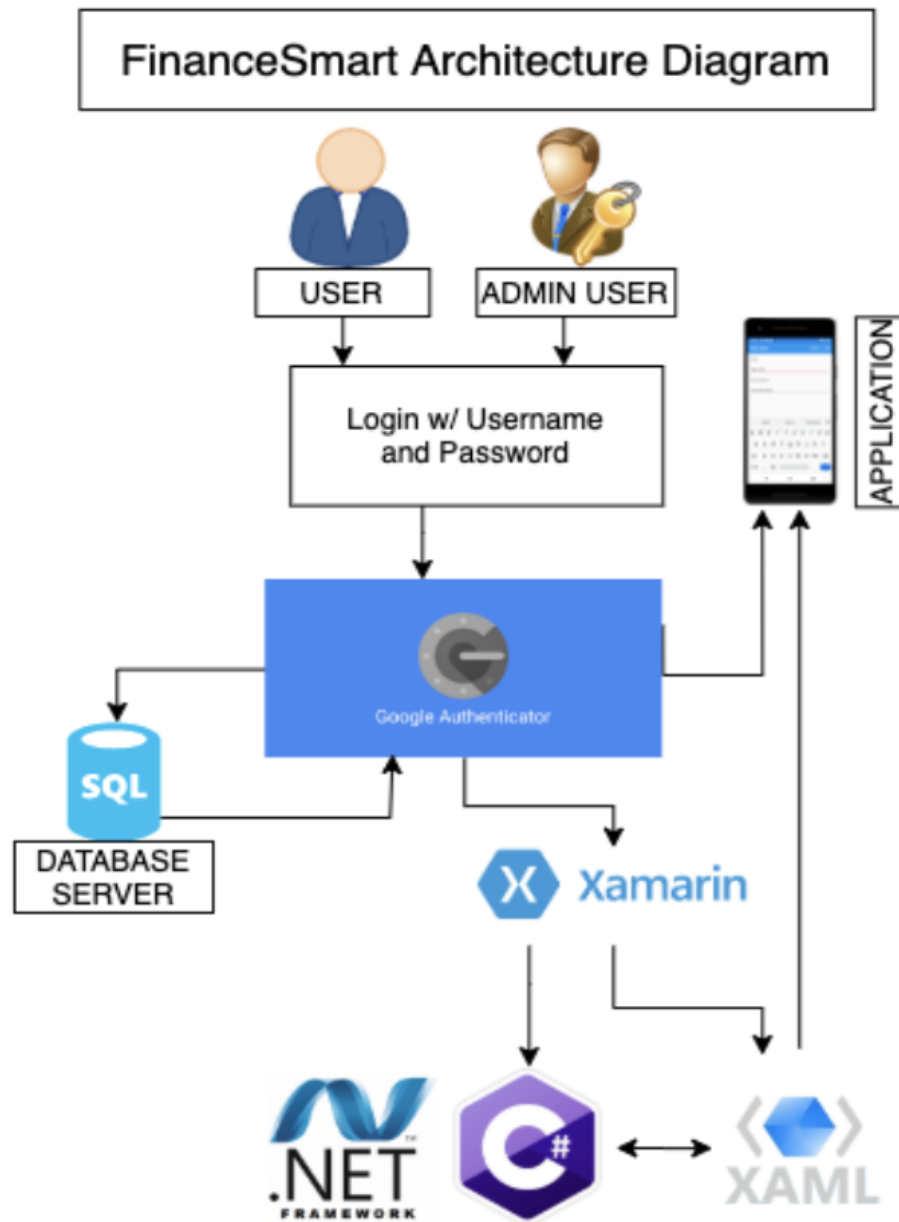


Figure 3 – Architecture diagram

Test Plan

Importance of Test Plan:

Test Plan is a very important aspect of any project. Test Plan helps not only the managers of the project but also the developers, clients and the users. It is like a guidebook which guides us throughout the projects so we can follow it and make changes and get the best result possible.

Overview, Product Details, and Users:

We will test our application by performing actions as a user and watching for output. Expected output will ensure the functionality of the application is in working state which is the main reason we will perform user actions and watch the result to see if the result comes as expectation. Finance smart is a Xamarin based application designed to help mainly young teenagers and late adolescents with financial success. The main goal of the app is to educate the users on financing, so they better understand their spending. Our user groups are depicted below in **Figure 4 – FinanceSmart Users**.

- (a) Young teenagers and late adolescents
- (b) Admin Users
- (c) Anyone

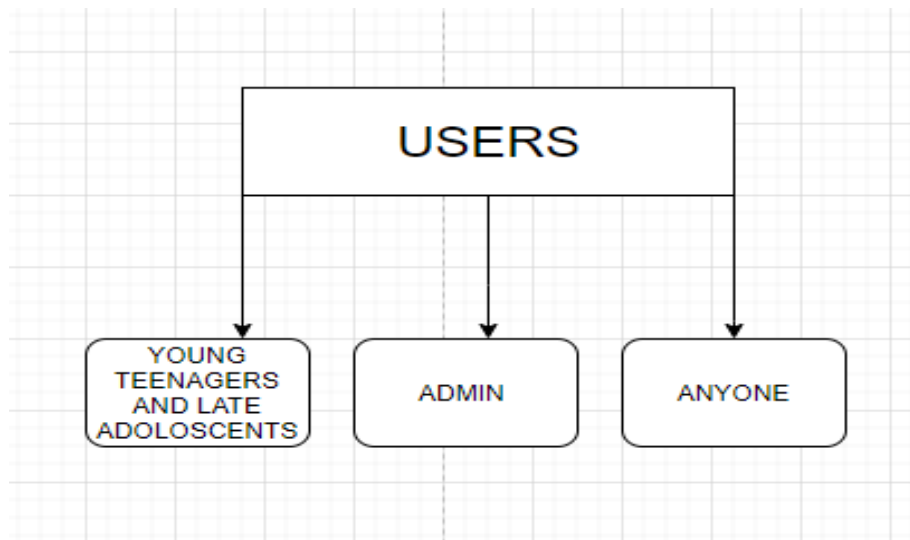


Figure 4 – FinanceSmart Users

Scope of Testing:

The goal of testing FinanceSmart is to find any bugs or errors in our application so that our users have the best experience of our application. In order to check that every function works it is really important we test the application thoroughly.

Testing and Reporting:

The testing of our application will have some scenarios. The scenarios are logging in, functionality of the features inside the application and successfully logging out. In any of these scenarios if the attempt is not successful then the issue must be documented and should be sent back for fixing the issues.

Assumptions - pass/fail:

All testing cases should pass, if any fails that needs to be fixed by the development team and should be documented for future references. The application should pass all the test cases to be successful. To use application features, assumption is the user is logged in to the application successfully.

Test Cases:

1- Login

Step 1: Put in username and password

Step 2: Click the login button

What we expect: User should be able to login successfully as long as they provide correct username and password. After logging in, the user can check out all the features and use them.

2- Add item

Step 1: Click Add button

What we expect: User should be able to click on 'Add' button to add items or accounts which have balances to pay and the item should show up on home screen when done.

3- Use a Feature

All the features are optional. A user may or may not have these accounts.

Step 1: Click on the feature - How to use Credit?

What we expect: The user should successfully be able to click on the feature and learn about how to use credit and able to go back to previous screen when done.

Step 2: Click on the feature - Budget Case

What we expect: The user should successfully be able to click on the feature, use the feature in order to create a monthly weekly or even daily budget and able to go back to previous screen when done.

Step 3: Click on the feature - Bank accounts

What we expect: The user should successfully be able to click on the feature and use the feature in order to link their bank accounts to the app safely and securely.

Step 4: Click on the feature - Credit card accounts

What we expect: The user should successfully be able to click on the feature and use the feature to view their credit card debt to track their expenses and able to go back to previous screen when done.

Step 5: Click on the feature - Personal loan accounts

What we expect: The user should successfully be able to click on the feature and use the feature to view their personal loan accounts with amount, check the payment dates and able to go back to previous screen when done.

Step 6: Click on the feature - Student loan accounts

What we expect: The user should successfully be able to click use the feature to view their student loan accounts, keep track of the payment due dates and able to go back to previous screen when done.

Step 7: Click on the feature - Payday accounts

What we expect: The user should successfully be able to click on the feature and use the feature in order to view their payday loan accounts and check the payment dates.

4- User Logout

Step 1: Logout

What we expect: The user should be able to logout successfully.

Future Recommendations/Recommendations for improvement:

If we were to start over the project all over again,

- We would definitely focus on our communication among the team.
- We would expect each of us to respond in a quick manner if not connect to advisors as soon as possible to eliminate possible future issues.
- We would be strict on completing defined tasks in work breakdown structure (WBS) in defined time.
- Also, we would like to have accessibility on submitting assignments to all the members in team.

Conclusion

Our multi - faceted application FinanceSmart will not only teach the user how to budget but it will also teach them how to use/build credit and track their spending. FinanceSmart is a Xamarin based mobile application targeted at middle to late adolescents to teach them how to be prepared for financial success. This will help middle to young adolescents understand their finances for adulthood.

Teamwork is a great place to start any project but each and every member needs to be obedient and honest towards the agreement. If any of the sign of future issue is gliding, don't delay escalating it to the leaders. Test plans are compulsory for the testing aspects which helps to fix bugs found in the earlier stage. During this project, we faced so may issues at the same time we were able to learn basics of Xamarin platform. It was great learning on the go. Most importantly, we learned the value of team and how a team member role is important for whole teams' success or failure.

Works Cited

Pascarella, Dani. "4 Stats That Reveal How Badly America Is Failing At Financial Literacy." *Forbes*, Forbes Magazine, 9 Apr. 2018, www.forbes.com/sites/danipascarella/2018/04/03/4-stats-that-reveal-how-badly-america-is-failing-at-financial-literacy/#8218e352bb7b.

Gabler, Neal. "The Secret Shame of Middle-Class Americans Living Paycheck to Paycheck."

The Atlantic, Atlantic Media Company, 11 July 2018,


<https://www.theatlantic.com/magazine/archive/2016/05/my-secret-shame/476415/>.

<https://blog.equifax.com/credit/financial-literacy/>

<https://www.investopedia.com/articles/pf/06/budgeting.asp>

Appendix

Appendix A – Poster



FinanceSmart

Team 43 – Jeavonte Staley, Devansh Patel, Shiv Joshi
Technical Advisor – Abdou Fall

College of Education, Criminal Justice, and Human Services – School of IT

About


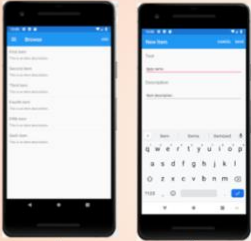
FinanceSmart is a Xamarin based mobile application targeted at middle to late adolescents in order to teach them how to prepare themselves for financial success, Credit Card debt and a lack of financial knowledge, to prepare young adults for adulthood.

Problem


Because of Credit Card debt and lack of financial knowledge to prepare young adults for adulthood, most adults do not have enough capital saved up to cover them in any financial emergency or are so deep into debt that they are just trying to pay it back but don't know how to effectively go about it. Also, currently available applications don't have customizable features that fits user needs.

Solution

FinanceSmart will be a cross - platform application with an easy to use intuitive user interface to entice adolescents to take responsibility of their finances and financial literacy. Customizable budgeting features to keep user up to date on financial aspect.



TECHNOLOGY



Conclusion and Statement

Our multi - faceted application FinanceSmart will not only teach the user how to budget but it will also teach them how to use/build credit and track their spending.