



LITAB

A life time financial assist application

Member

Bowen Zhang s3617571

Cheng Chen s3728207

Bowen Yang s3665803

Zongming Liu s3713771

Lei Guo s3516439

Zheshun Shi s3696260

Assignment 2

Team Member	2
Project Description	4
Project prototype	4
Core Features	5
Core Feature 1	5
Core Feature 2	6
Core Feature 3	7
Core Feature 4	8
Core Feature 5	9
Core Feature 6	10
Extended Features	11
Extended Feature 1	11
Extended Feature 2	13
Extended Feature 3	14
Project Estimation	15
Technology Listing	27
Collaborative Workspaces	27
Software	27
Tools	29
Resources	30

Team Member

Bowen Zhang s3617571

Background & Passion - Basic coding skills and passionate about games and Artificial Intelligence.

Skill & Interest - Good at Coding, Learning and organizing and interested in all sorts of games and AI or advanced technologies.

Weakness – I'm not good at English writing.

Role – Organiser & programmer

Cheng Chen s3728207

Background & Passion - Had basic coding experience with VS, Pascal, Python and Java. Strong interest in building a software.

Skill & Interest - Enthusiast of hardware and game building, Good at organising. Interested in coding and function developing.

Weakness - Weak at visual design.

Role - Mainly working on software structures and feature design.

Bowen Yang s3665803

Background & Passion - I have basic experience with Java programming language and program interface design.

Skill & Interest - Better at text editing than coding, but I am working hard to learn programming and are interested in the IT words.

Weakness - Probably the coding skills need to improve.

Role - I expect to become a competent team member that be able to do a good job of assigning tasks and be able to present their own views on the project.

Zongming Liu s3713771

Background & Passion - I know some basic coding skills and static web layout coding skills.

Skill & Interest - I am good at designing the website

Weakness - weak at coding

Role - design the UI and interface of the app

Lei Guo s3516439

Background & Passion - I have basic coding skills in Java, .net and web programming.

Skill & Interest - Good at teamwork and interesting in computer games and travelling

Weakness - Weak at English

Role - Good team member

Zheshun Shi s3696260

Background & Passion - basic Java and web programming skills.

Skill & Interest - interested in VR.

Weakness - weak at English grammar and oral speaking.

Role - perfect teammate.

Project Description

Our team is building a mobile application called Litab which is a combination of Tally book and Reminder. The main purpose of this application is to keeping track on personal expense and to remind user what's coming next and what they want to buy or do in the future. Users can plan their future expense based on their monthly average spending and income and users can categorize their expense for better management. The application will also tell the user if their plan have fulfilled or short by how much as well, remind them if they need to cut their budget or create extra income to achieve their goal. Finally, this app will feature all the basic function a tally book has such as list all consumption, categories and a clean and friendly interface.

Project prototype

InvisionApp: https://invis.io/HGO1CBQCJSQ#/319736797_1-1-Home

Core Features

Core Feature 1

The image displays three screenshots of a mobile application named 'Tally'. Each screenshot shows a different time period and category breakdown.

View	Period	Category	Amount
Daily	24 Aug 2018	Cost	- \$ 5.4
		Income	\$ 10.0
		Yesterday	\$ 4.6
		Yesterday	24 Aug 2018
Monthly	July 2018	Cost	- \$ 9021.50
		Income	\$ 6020.00
		Yesterday	- \$ 3001.50
		Yesterday	24 Aug 2018
Yearly	2018	Cost	- \$ 34550.00
		Income	\$ 50000.00
		Yesterday	- \$ 3001.50
		Yesterday	24 Aug 2018

Additional data from the screenshots:

- Daily View (24 Aug 2018):** Coffee (- \$ 5.40), Dinner (- \$ 11.50), Lunch (- \$ 21.50), Salary (\$ 4000.00), Total (\$ 3961.60).
- Monthly View (July 2018):** Cost (- \$ 3120.00), Income (\$ 5900.10), Total (\$ 2780.10).
- Yearly View (2018):** Cost (- \$ 34550.00), Income (\$ 50000.00), Total (\$ 15450.00).

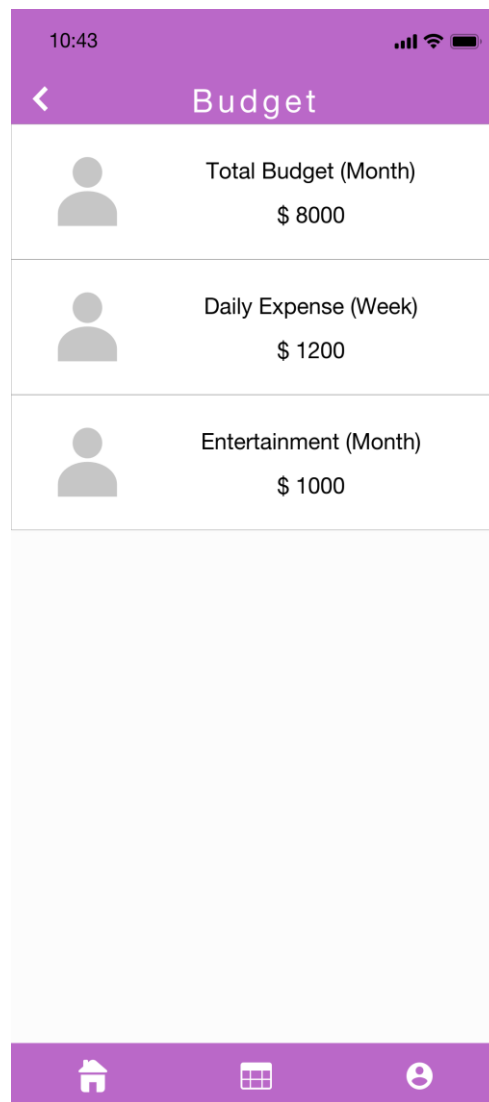
Core Feature 1 Validation Testing:

Check the income and cost categories. The users can check the categories by daily, monthly or yearly. The app will calculate the statement. They also can click each category to change the details. If the app prompts for valid categories, we can make sure that every data is recorded into the correct category.

Author: Bowen Zhang

Create Date: {26/08/2018 }

Core Feature 2



Core Feature 2 Validation Testing:

Set the budget for weekly and record some data to perform overspend. The users can also set the Daily expense, Entertainment budget or other budgets. If the app shows an alarm when the current spending value exceeds the budget set, this function is working properly.

Author: { Cheng Chen }

Create Date: { 26 /08/2018 }

Core Feature 3

The screenshot shows a mobile app interface titled 'Routine'. At the top, there's a status bar with the time '11:01' and signal/battery icons. Below the title bar, there's a 'Start Date' section with a table listing three dates:

1	January	2016
2	Feburary	2017
3	March	2018

Below the table are two input fields: 'Name' and 'Amount'. Under these fields are two tabs: 'Cost' (selected) and 'Income'. Below the tabs are three frequency options: 'Daily' (selected), 'Weekly', and 'Monthly'. At the bottom of the form is a 'Save' button. The app has a purple theme and a bottom navigation bar with icons for home, calendar, and profile.

Core Feature 3 Validation Testing:

Input the name and the money in the category creation menu. The users can choose the cost section or income section. They also can choose the cycle time. The app will record the data. If they choose the daily option, the app will put the data into tally everyday. If the category shows up on the spending input page, this function is working properly.

Author: { Bowen Zhang }

Create Date: { 26/08/2018 }

Core Feature 4

10:59

< Goals

Name of your goal
Placeholder

Amount
Placeholder

Description
Placeholder

Priority >

When By Date Estimate
Placeholder

Home Calendar Profile

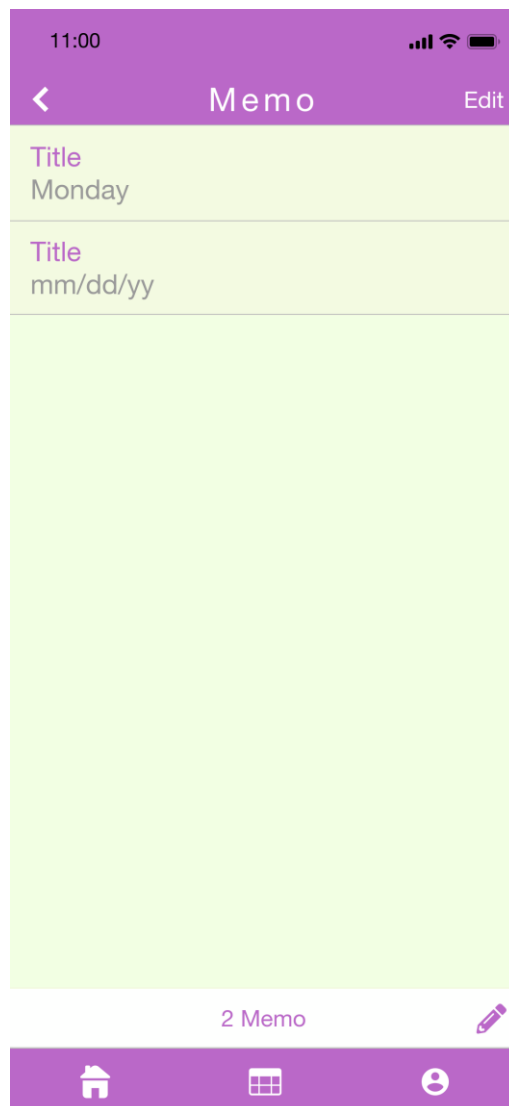
Core Feature 4 Validation Testing:

Set a target in money box with preferred purchase date. If the users choose the “By date”, the users can set the date and the app will calculate the money the users need to save everyday. If the users choose the “Estimate”, the app will calculate the time by the expense history that users will know how many days they need to wait. Users can see the details of all the things. If the app correctly shows the amount of money needed to be saved each week, this function is working properly.

Author: { Bowen Zhang }
{ 26/08/2018 }

Create Date:

Core Feature 5



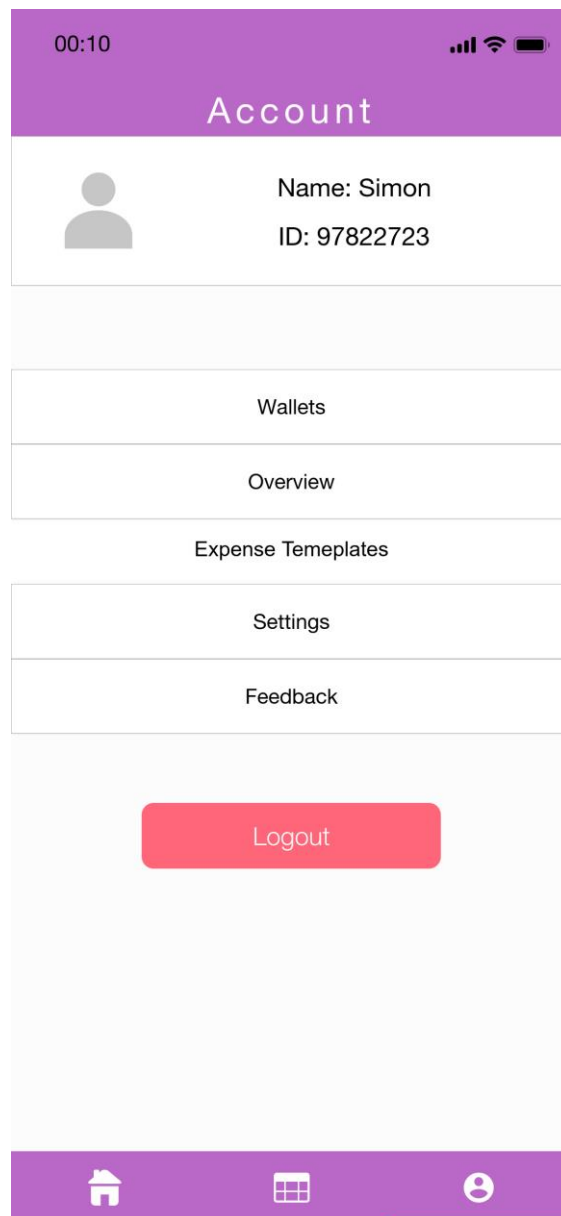
Core Feature 5 Validation Testing:

Check all the notes and reminders. The users can also click the pencil icon or click the “edit” to change the memos. If the notes are correctly displayed, the notes function is working properly. If the alarm goes off at the desired time, the reminder function is working properly.

Author: { Bowen Zhang }

Create Date: { 26/08/2018 }

Core Feature 6



Core Feature 6 Validation Testing:

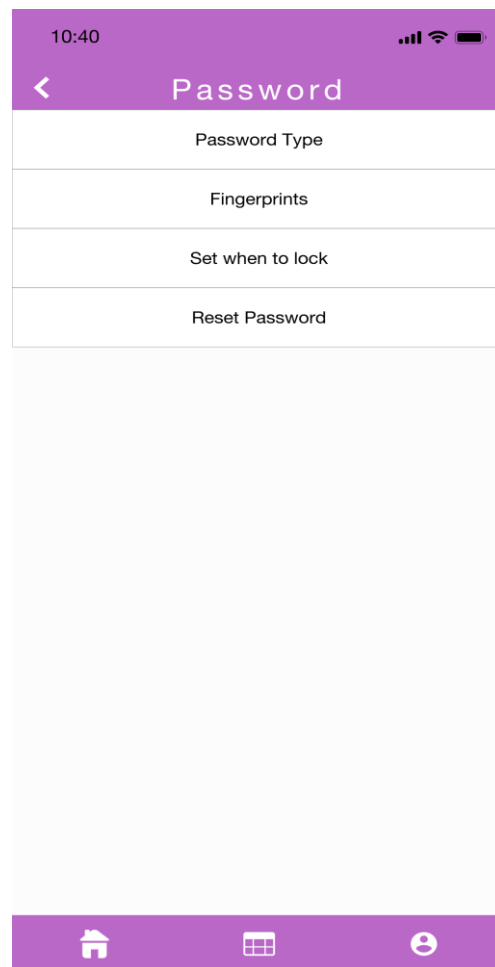
The account setting allowed users select to use different functions. The users can click the buttons to do check the details.If the users can click different buttons and go to the functions, the setting function is working properly.

Author: { Bowen Zhang }

Create Date: { 26/08/2018 }

Extended Features

Extended Feature 1



Extended Feature 1 Validation Testing:

Input a valid username and password. If successfully logged in, it is working properly. Input an invalid username or password. If the login failed, the login function is fully working.

Author: {Cheng Chen }

Create Date: { 07/09/2018 }

Extended Feature 2



Extended Feature 2 Validation Testing:

Set a template and test with some text messages corresponding to the template. If the app recognizes the text message and records the correct value in the text, this function is working properly.

Author: {Cheng Chen }

Create Date: { 07/09/2018 }

Extended Feature 3



Extended Feature 3 Validation Testing:

For different users from different countries can use the app. If the users can choose the language they use, this function is working properly.

Author: {Cheng Chen }

Create Date: { 07/09/2018}

Project Estimation

	Excited Birds							
	LITAB							
{Task Name}	Trevor	Cheng Chen	Bowen Zhang	Zong ming Liu	Lei Guo	Bowen Yang	Zhesh un Shi	{Aver age Hour}
MVF1 Set income and cost categories	27.5	25.0	25.0	26.0	20.0	26.0	28.0	25.0
MVF2 Set the budget	27.9	26.0	20.0	27.0	30.0	24.0	25.0	25.3
MVF3 Record the income and spending	26.8	27.0	19.0	26.0	20.0	24.0	30.0	24.3
MVF4 Set a target in money box	37.8	35.0	34.0	34.0	32.0	35.0	36.0	34.3
MVF5 Set the notes and reminders	28.2	26.0	20.0	24.0	30.0	28.0	26.0	25.7
MVF 6 Account setting	26.6	26.0	20.0	23.0	24.0	25.0	27.0	24.2
EVF1 Login	25.3	28.0	15.0	23.0	26.0	22.0	24.0	23.0
EVF2 Identify SMS	35.2	30.0	28.0	32.0	30.0	34.0	38.0	32.0
EVF3 Language setting	29.0	29.0	20.0	28.0	25.0	30.0	26.0	26.3
Totals:	264.2	252.0	201.0	243.0	237.0	248.0	260.0	240.2

For this project, the biggest challenge for us is that we don't have mobile application programming experience, we have not studied mobile application programming, so we need to spend a lot of time to learn. This means that our estimated time will be very high. For assignment 2, Bowen Zhang, Cheng Chen and Zheshun Shi will work on design. Bowen Yang, Lei Guo and Zongming Liu will work on report. Everyone will work on coding. Although everyone could have different task, we will discuss every step of this project, everyone can share their ideas and comments.

Lei Guo

For this assignment, my main task is report, so I will not spend a lot of time on design, but I will spend some time to communicate with team member about design. I will focus on coding, in order to develop this app, I have to spend a lot of time to learn related knowledge.

MVF 1:

I need to spend 4 hours to learn and practice mobile application programming, 3 hours to research mobile application programming and application development process, 2 hours to design, coding will be spent the most time, it could be 10 hours because I have to try many times to develop this feature. Testing will be spent 1 hour. I estimate it takes a total of 20 hours.

MVF 2:

Like MVF 1, I need to spend 4 hours to learn and practice mobile application programming, 3 hours to research mobile application programming and application development process. For design, it could be spent 3 hours. Coding will be spent 19 hours. Testing will be spent 1 hour. I estimate it takes a total of 30 hours.

MVF 3:

Each week, I need to spend 4 hours to learn and practice mobile application programming, 3 hours to research mobile application programming and application development process. Design will need 2 hours, coding will be spent 10 hours. Testing will be spent 1 hour. I estimate it takes a total of 20 hours.

MVF 4:

For me, this feature is the most challenge, I need to spend 5 hours to learn and practice mobile application programming, 5 hours to research mobile application programming and application development process, 3 hours to design, coding will be spent the most time, it could be 17 hours. Testing will be spent 2 hours. I estimate it takes a total of 32 hours.

MVF 5:

The same as MVF 4, I need to spend 5 hours to learn and practice mobile application programming, 5 hours to research mobile application programming and application development process, 2 hours to design, coding will be spent 17 hours. Testing will be spent 1 hour. I estimate it takes a total of 30 hours.

MVF 6:

This feature is connected to other functions. Therefore, I need to spend a lot of work to code and testing, so that there are no bugs and works well. I will spend 5 hours on learning, 5 hours on research, 2 hours on design, 10 hours on coding and 2 hours on testing.

EVF 1:

Spending 4 hours to learn and practice mobile application programming, 4 hours to research mobile application programming and application development process, 2 hours to design, 15 hours to code and 1 hour to test. I estimate it takes a total of 26 hours.

EVF 2:

Like MVF 4, this feature is also very challenging for me. So I need to spend 5 hours to learn and practice mobile application programming, 5 hours to research mobile application programming and application development process, 3 hours to design, 15 hours to code and 2 hour to test. I estimate it takes a total of 30 hours.

EVF 3:

This feature is not easy. It is my first time to develop language setting. I need to spend 4 hours to learn and practice mobile application programming, 4 hours to research mobile application programming and application development process, 2 hours to design, coding will be spent 14 hours. Testing will be spent 1 hour. I estimate it takes a total of 25 hours.

Bowen Zhang

For assignment 2, my main task is design. I will spend more time on design. No doubt, the basic features are most important. Once user open the app, what they saw and use at the first time is essential to keep them using the app. and these basic features is also what we promised to our user. that's why I would like to spend more time on it. for extensional features, it is not the reason why user choose to use our app at the first time, but what we tried to make our app better, it means users can be more tolerant if it does not work so well and we can fix it after without losing many users. Also, the design of the basic features determine the whole design style which basically cannot be changed again, so it is reasonable to spend more time and thoughts on it.

MVF 1:

Learning = 3 hours

Research = 4 hours

Design = 12 hours

Coding= 5 hours

Testing= 1 hour

This feature is display on our home screen which means it is like a face of our app and for the consistency, all the other section will be follow its style. and coding for this part is not that difficult. therefore, I would like to spend more time considering design style.

MVF 2:

Learning = 3 hours

Research = 5 hours

Design = 5 hours

Coding= 6 hours

Testing= 1 hour

This part is not hard on design because the style follows MVF 1, and coding is not hard either. but this is a feature that to tell user why our app is different. so every parts of it needs equally consideration.

MVF 3:

Learning = 2 hours

Research = 4 hours

Design = 2 hours

Coding= 10 hours

Testing= 1 hour

This is a part that we think is convenient to users. the idea is simple, user can easily record their routine without input every time. the coding part can be tricky because the data will inject to MVF 1 but cannot affect any change that user might do. bugs is most intolerance. so it is necessary to spend more on coding.

MVF 4:

Learning = 3 hours

Research = 6 hours

Design = 6 hours

Coding= 18 hours

Testing= 1 hour

This feature is what we tried to make our app better and different than other similar app. so I would spend more time on it. same as MVF3, coding is bugs intolerance. the app needs to quickly calculate the date or the money user need to achieve their goal. and daily spend or other financial issues can easily affect that.

MVF 5:

Learning = 3 hours

Research = 8 hours

Design = 4 hours

Coding= 4 hours

Testing= 1 hour

This part is not easy to make any difference and coding is not hard. so I would not spend too many time on it. but it is basic feature. so I would like to do more research and try to make it better.

MVF 6:

Learning = 2 hours

Research = 2 hours

Design = 5 hours

Coding= 10 hours

Testing= 1 hour

this part is more about hard working because too many functions in this part, the design can follow the others with some detailed adjustment but the coding needs more work, like quick switch, no bugs, no delay etc.

EVF 1:

Learning = 0 hours

Research = 3 hours

Design = 6 hours

Coding= 5 hours

Testing= 1 hour

This part seems very easy, and it is but it is also hard to make an easy one better and log in more smoothly.

EVF 2:

Learning = 7 hours

Research = 6 hours

Design = 1 hours

Coding= 13 hours

Testing= 1 hour

This part could be the most difficult one because how to connect to message and how to recognize the financial message specifically without delay, error or miss needs very careful thought and good coding technique. there is basically no design required.

EVF 3:

Learning = 1 hours

Research = 12 hours

Design = 4 hours

Coding= 2 hours

Testing= 1 hour

Honestly, this part need not much design required except how the font-style fit in the app. translation is the only tricky part.

Bowen Yang

In assignment 2, I will concentrate on writing report and doing some programming code.

MVF 1:

This feature is the display on our home screen which it is not difficult. However, I focus on writing the report. Therefore, I will spend a little time on the learning section and research section.

Spending 5 hours on learning

Spending 5 hours on research

Spending 2 hours on design

Spending 13 hours on coding

Spending 1 hour on test

MVF 2:

The app will allow users to save several budget settings to control their spending. For coding, it's easier than other features. I will spend more time to the learning section and research section.

Spending 4 hours on learning

Spending 4 hours on research

Spending 2 hours on design

Spending 12 hours on coding

Spending 1 hour on test

MVF 3:

This part of the code can be difficult because the data is injected into MVF 1. At the same time, it is necessary to spend more money on testing.

Spending 4 hours on learning

Spending 4 hours on research

Spending 2 hours on design

Spending 12 hours on coding

Spending 1 hour on test

MVF 4:

This part is quite difficult. Applications should not simply perform calculations to get results. I need to process the code to make the application more efficient to ensure the use of the feature. Hence, I will spend more time in the code section.

Spending 6 hours on learning

Spending 5 hours on research

Spending 3 hours on design

Spending 15 hours on coding

Spending 1 hour on test

MVF 5:

This is the main function in addition to other major features. Design is not my main job. My work is mainly about code and collecting information.

Spending 5 hours on learning

Spending 4 hours on research

Spending 2 hours on design

Spending 14 hours on coding

Spending 1 hour on test

MVF 6:

Design is not my main job. My work is mainly about code and collecting information. Hence, I will focus on the learning section in addition to the research section.

Spending 5 hours on learning

Spending 4 hours on research

Spending 2 hours on design

Spending 13 hours on coding

Spending 1 hour on test

EVF 1:

This part is not difficult and I will spend more time on the coding section.

Spending 5 hours on learning

Spending 4 hours on research

Spending 2 hours on design

Spending 10 hours on coding

Spending 1 hour on test

EVF 2:

This feature could be the most challenging part of the app, which is our project should be smart. I will spend more time to program it and make sure it can run smoothly.

Spending 5 hours on learning

Spending 5 hours on research

Spending 2 hours on design

Spending 21 hours on coding

Spending 1 hour on test

EVF 3:

For the language function, we might need help from several language specialists. Therefore, I need more time to collect more resources in the research section.

Spending 7 hours on learning

Spending 6 hours on research

Spending 2 hours on design

Spending 14 hours on coding

Spending 1 hour on test

Cheng Chen**MVF 1:**

This feature is the first one for the structure of the whole app. It's not a difficult one but we have to get familiar with the overall program structure and some further edit on the interface design.

Spending 4 hours on learning

Spending 4 hours on research

Spending 4 hours on design

Spending 12 hours on coding

Spending 1 hour on test

MVF 2:

The app will be able to let users save several budget settings to control their spending. In terms of coding, it's relevantly easier than other features. I will spend more time testing to make sure nothing wrong happens in this section.

Spending 5 hours on learning

Spending 5 hours on research

Spending 4 hours on design

Spending 12 hours on coding

Spending 1 hour on test

MVF 3:

This is the main feature of the app. I have to be very careful on this part since any small mistake will crash the app. This section include lots of user input so huge amount of time will be spent for testing this feature to make sure everything runs smoothly.

Spending 6 hours on learning

Spending 4 hours on research

Spending 5 hours on design

Spending 11 hours on coding

Spending 1 hour on test

MVF 4:

This part is quite challenging. The app should not simply do a calculation to get the result since this would lead to strange outputs. I need to work on the codes to make the app smarter. Making sure that the app is showing readable results.

Spending 7 hours on learning

Spending 8 hours on research

Spending 4 hours on design

Spending 15 hours on coding

Spending 1 hour on test

MVF 5:

This is a main feature that is apart from other main features. It is a relatively simple feature to implement. I will be mainly working on the codes to make it more versatile.

Spending 5 hours on learning

Spending 4 hours on research

Spending 6 hours on design

Spending 10 hours on coding

Spending 1 hour on test

MVF 6:

This is a main feature that is apart from other main features. It is a relatively simple feature to implement. I will be mainly working on the codes to make it more versatile.

Spending 5 hours on learning

Spending 4 hours on research

Spending 6 hours on design

Spending 10 hours on coding

Spending 1 hour on test

EVF 1:

Currently I'm not sure how login works on most apps. I will spend more time researching on this part. Maybe we have to get a server to make everything work. Optimized codes are also very important.

Spending 8 hours on learning

Spending 8 hours on research

Spending 1 hours on design

Spending 10 hours on coding

Spending 1 hour on test

EVF 2:

This feature is very challenging for us to implement. The app should be smart and do not only recognize fixed templates. Otherwise the recognition might be unstable if the text message changes. Lots of testing will be put into this feature.

Spending 5 hours on learning

Spending 5 hours on research

Spending 2 hours on design

Spending 10 hours on coding

Spending 8 hour on test

EVF 3:

As for language, we might need help from several language specialists. Research is also a way for us to get over it. We will make sure the translations are accurate and readable.

Spending 8 hours on learning

Spending 7 hours on research

Spending 5 hours on design

Spending 8 hours on coding

Spending 1 hour on test

Zongming Liu

In this assignment 2, my work is to describe the core features and extended features in report. When I writing the description of the features, I also will check the design of the app. If I find some mistakes or not humanized design, I will have a discussion with the design to improve the app.

MVF 1:

Learning = 5 hours

Research = 4 hours

Design = 2 hours

Coding= 13 hours

Testing= 2 hours

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

MVF 2:

Learning = 5 hours

Research = 5 hours

Design = 2 hours

Coding= 13 hours

Testing= 2 hours

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

MVF 3:

Learning = 5 hours

Research = 5 hours

Design = 3 hours

Coding= 13 hours

Testing= 1 hour

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

MVF 4:

Learning = 7 hours

Research = 7 hours

Design = 4 hours

Coding= 15 hours

Testing= 1 hour

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

MVF 5:

Learning = 5 hours

Research = 4 hours

Design = 2 hours

Coding= 12 hours

Testing= 1 hour

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

MVF 6:

Learning = 7 hours

Research = 6 hours

Design = 6 hours

Coding= 2 hours

Testing= 2 hours

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

EVF 1:

Learning = 4 hours

Research = 4 hours

Design = 2 hours

Coding= 13 hours

Testing= 1 hour

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

EVF 2:

Learning = 7 hours

Research = 6 hours

Design = 3 hours

Coding= 18 hours

Testing= 4 hours

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

EVF 3:

Learning = 6 hours

Research = 5 hours

Design = 2 hours

Coding= 12 hours

Testing= 3 hours

I have talked with the design team and to learn the features. I also learn how to use the features by design team help. At the same time, we have the team meeting for the app design. I spend most of time to write the description on the app and check the design.

Zheshun Shi

In assignment 2, I focus on designing application which discuss how it would be easier use for users and how it be more aesthetic in general. Thus, I will spend a lot of time on design. It is obvious that UI and UX are very important for users.

MVF 1:

Learning = 6 hours

Research = 6 hours

Design = 5 hours

Coding= 10 hours

Testing= 1 hour

MVF 2:

Learning = 5 hours

Research = 5 hours

Design = 3 hours
Coding= 11 hours
Testing= 1 hour

MVF 3:

Learning = 6 hours
Research = 7 hours
Design = 5 hours
Coding= 11 hours
Testing= 1 hour

MVF 4:

Learning = 8 hours
Research = 7 hours
Design = 6 hours
Coding= 14 hours
Testing= 1 hour

This feature is very difficult for me, I will spend the most time on this feature.

MVF 5:

Learning = 6 hours
Research = 5 hours
Design = 4 hours
Coding= 10 hours
Testing= 1 hour

MVF 6:

Learning = 5 hours
Research = 5 hours
Design = 3 hours
Coding= 13 hours
Testing= 1 hour

EVF 1:

Learning = 4 hours
Research = 3 hours
Design = 4 hours
Coding= 12 hours
Testing= 1 hour

This feature is not very difficult, I will focus on design and coding.

EVF 2:

Learning = 8 hours
Research = 7 hours
Design = 5 hours
Coding= 17 hours
Testing= 1 hour

The same as MVF 4, I have to spend a lot of time on this feature.

EVF 3:

Learning = 5 hours

Research = 4 hours

Design = 4 hours

Coding= 12 hours

Testing= 1 hour

This feature is very important, so I have to design well and coding well.

Technology Listing

Collaborative Workspaces

1. Google doc

<https://drive.google.com/drive/u/0/folders/1CsO1w9nZAClBlzIfCf-IPmvl-VHngsY->

Google doc is the tool our team uses to write reports. Controlled by our tutor and given the permission of our team members to edit.

2. Github

<https://github.com/s3665803/excited-bird>

Github is controlled by our team member Bowen Yang and gives us team members permission to edit.

3. Trello

<https://trello.com/b/HchjnuLb/tr-excited-birds>

Trello is controlled by our tutor and given the permission of our team members to edit, which can show the progress of our assignment.

Software

Wechat

a. Short description

WeChat users can share text and pictures with friends through the smartphone client, and support smart phone chat software for group chat and voice and video intercom functions.

b. Rationale

In order to be able to respond in time and propose ideas after class, we create a wechat group, which it can help our group eliminate misunderstandings and improve efficiency.

Basically, six members of our team use WeChat every day to discuss assignments and to check and correct the completed parts.

The connection between each part is powerful despite the fact that each member has there individual parts., in order to guarantee the quality of the assignment, we need to report on the task.

c. Specific version

ios 6.7.2 For free.

App Icon Gear

a. Short description

App Icon Gear is a tool for Mac, iOS developers who want to efficiently make App Icons, Launch Images and Image assets. Gear works perfectly with Xcode.

b. Rationale

It's very convenient to work with Xcode and very simple to operate, which it can quickly suppress pictures of various sizes. We can drag two images at the same time (one horizontal and the other vertical), and Gear will suppress the omnidirectional Launch Image. Therefore, our group select this app to design various icons for our project.

In addition, currently supported a plenty of types and these beautiful icons are easy to identify.

12 kinds of iOS App Icon (6 iPhones, 6 iPads)

iOS Launch Image 20 kinds

MAC App Icon 10 kinds

Apple Watch App Icon 8 kinds

Xcode icon set (XC Asset) 3 kinds

c. Specific version

1.5.5 For free.

ADOBE PHOTOSHOP CC

a. Short description

Photoshop is one of Adobe's most famous image processing software, which integrates image scanning, editing and editing, image creation, advertising creativity, image input and output into one image processing software.

b. Rationale

Photoshop is the most powerful drawing software available today. Photoshop's tools are so smart, which we can use the pens, filters and modes that come with the software to achieve all the effects you can think without any extension. In addition, powerful expandability allows us to perform tasks in a wide variety of pen filters in a wide variety of pen filters. Therefore, we discuss use the Photoshop to design background and logo of our application.

c. Specific version

V13.0.3

First month is free.

A\$39.59/mo (incl. GST)

Tools

1. Xcode

a. Short description

Xcode is an integrated development environment (IDE) provided by Apple to developers for the development of MacOS, iOS, WatchOS and tvOS applications.

b. Rationale

Because our group will develop a program on the ios platform. Xcode has a unified user interface design, coding, and testing that is the fastest way to develop OS X and iOS applications. Support multi-language programming programs C, C++, Objective-C or Java programming and it can publish the program directly to the App Store. Therefore, the Xcode is the most suitable IDE for our program.

c. Specific version

Xcode 9.4.1 For free.

d. Alternative plan

Eclipse

<https://www.eclipse.org/>

2. proto.io

a. Short description

Proto.io is a dedicated mobile phone prototyping platform that can build and deploy prototypes of fully interactive mobile applications and simulate similar finished products.

b. Rationale

It's convenient for quickly creating an interface because it has a rich set of standard UI elements, and many of them are customizable. In addition, it can be many screens in a project, and transitions can be made between these screens.

c. Specific version

Proto.io 6 Free for 15 days.

d. Alternative plan

invision

<https://www.invisionapp.com/>

Resources

Github

GitHub is the world's largest code resource hosting sharing site with a lot of resources to learn to help us better complete our application.

<https://github.com/ilisan/dcf-d-mw-applet/blob/420c18ade379d07547652feac0d39d4a0ec28ae7/dss/eid-dss-admin-portal-control/src/main/java/be/fedict/eid/dss/admin/portal/control/bean/AccountingBean.java>

To start with, we did some research on the github in order to find some helpful resources, which can assist us to complete the program better before we start create our project. Beside, we selected several open source and conducted a group discussion. We discussed the usability and fluency of various application resources. Finally, the results of the discussion and the ideas generated have helped us greatly during the development process.

2. Spendee

Spendee is a gorgeous, user-friendly app that people love, which is the simplest way to control your finances. A successful project is a good example for developers. An application that can successfully test the consumer's test in the market for a long time can help us to avoid weaknesses.

Cleevio, viewed 1 Sep 2018, <<https://cleevio.com/spendee>>.

