

Hobby Checklist App

BY: AUSTIN CHRISTMAN

PROFESSOR: PAUL LIN

ADVISOR: LUO

DATE: 4/29/16



Outline

- ▶ Executive Summary
- ▶ Introduction
- ▶ Problem Statement and Solution
- ▶ System Requirements
- ▶ System Analysis
- ▶ System Design
- ▶ System Integration and Testing
- ▶ Validation
- ▶ Conclusion




Executive Summary

- ▶ Purpose of Project: To provide users with an app that will organize activities they would do in their free time and help keep track of progress.
- ▶ Project Time Period: Started Fall 2015 to April 29, 2016
- ▶ Deliverables:
 - ▶ Prototype: A working version of the app
 - ▶ Report: Summarizes all the aspects of the Design and Testing of the project
 - ▶ Presentation: Present the workings of the project to the audience

Introduction

- ▶ Part of App Development is to add value to the user's life
- ▶ Utility apps and entrainment apps both do this
- ▶ People want items that make their lives easier
- ▶ Lists help people remember things they are trying to keep track of.

Problem Statement

- ▶ Personal Problem: I was into more activities than I could keep track of.
 - ▶ Start one series before finishing the other, and end up forgetting where I left off.
 - ▶ People with similar interests had the same issue.
- 

Problem Solution

- ▶ Solution: Create an app that would organize these activities for people with similar interests.
- ▶ Divide it up by category, with a list for each individual category.
- ▶ Categories to include Anime, Manga, Comics, Books, TV, Video Games, Movies, and Miscellaneous.
- ▶ Keep track of progress on these activities (Pages read vs. Total Pages)

Inspiration


- ▶ Idea for this app was inspired by a website known as MyAnimeList
- ▶ MyAnimeList provides users with the ability to make lists for Anime and Manga.
- ▶ They provide the user with all the information for any entries (# of episodes, last update, user reviews)
- ▶ Different from my app since the user must provide all that information.

System Requirements

8

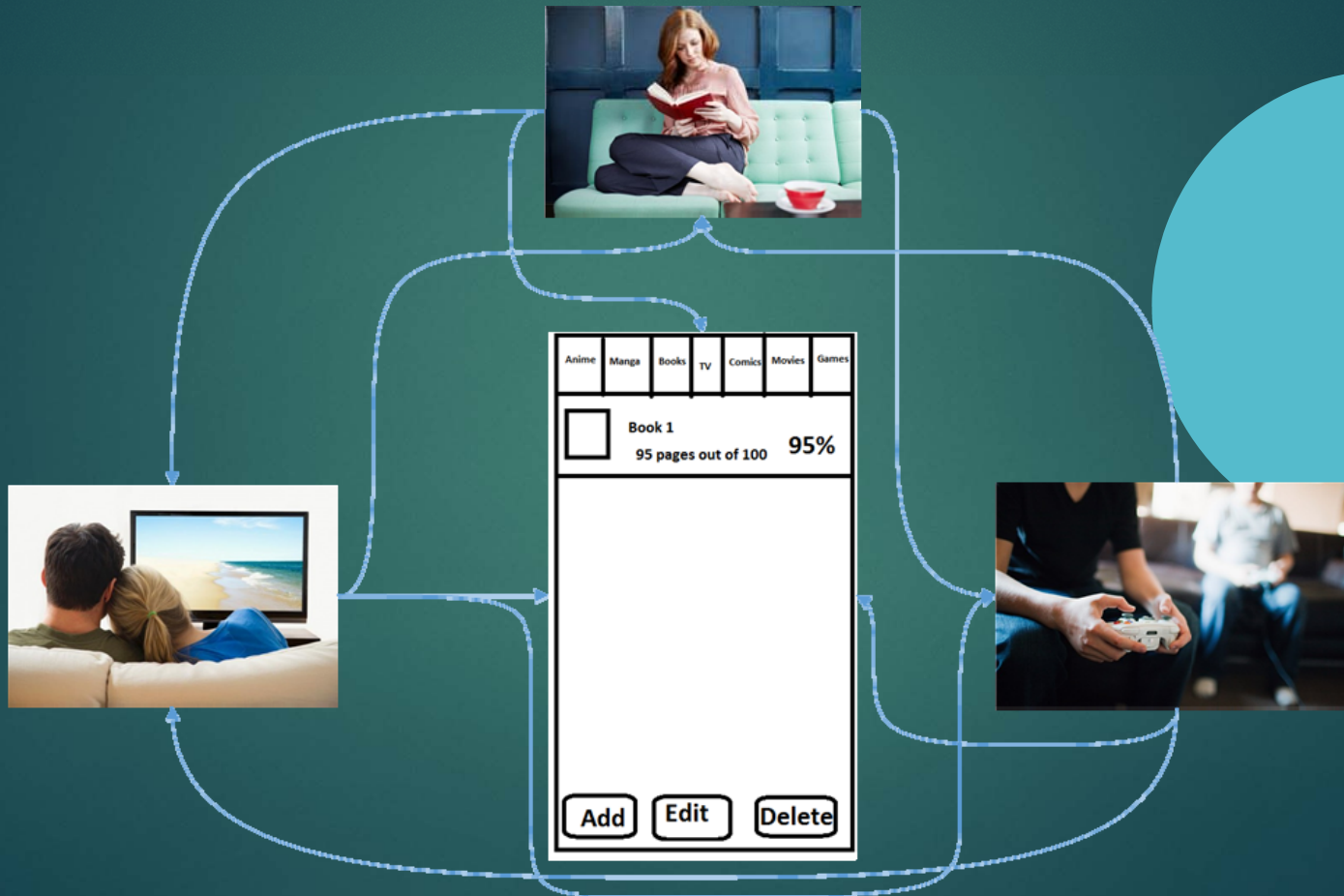
Project: Hobby Checklist App		
Revision: 2		
Date: 27-Nov-15		
Requirement Data		
ID	Requirement Type	Requirement (Shall or Should statements)
1	Functional	The application shall provide a user interface
2	Operational	The application shall keep track of the user's progress on a set of leisure activities
3	Operational	The application shall send out daily notifications using the phone's vibration function that will leave a text message to the user.
4	Functional	The application shall store data on the phone's storage
5	Functional	The project shall organize activities into categories
6	Performance	The categories shall be Anime, Manga, Comics, TV, Movies, Books, Video Games, and Miscellaneous
7	Functional	The project shall use the phone's vibration function for notifications.
8	Functional	The application shall use the phone's timer to know when to send daily notification
9	Performance	The application shall display progress completed up to the nearest % value based on the completion of the activity (ex. 95 pages out of 100 pages would be 95%)
10	Performance	The application shall send out the daily notification at noon local time +/- 5 minutes
11	Environmental	The application shall work on Android OS 4.4.2 on Samsung S3
12	Environmental	The application should work on Android OS 4.4.3 and later
13	Physical	The project shall reserve 300 MB for storage use
14	Physical	The project shall not use more than 500 MB of CPU
15	Physical	The application shall not be larger than 700 MB
16	Functional	The daily notification time should be user-specified
17	Environmental	The application shall be developed under the Android Studio programming environment
18	Functional	The user shall not be able to edit the category names
19	Functional	The application shall retrieve data off the phone's storage

System Requirements: Key Requirements

- ▶ The application shall keep track of the user's progress on a set of leisure activities.
 - ▶ The project shall organize activities into categories (Anime, Manga, Movies, TV, Comics, Books, Video Games, and Miscellaneous).
 - ▶ The application shall work on Android OS 4.4.2 or later
- 

System Analysis: OV-1

10



System Analysis: Early User Interface Design

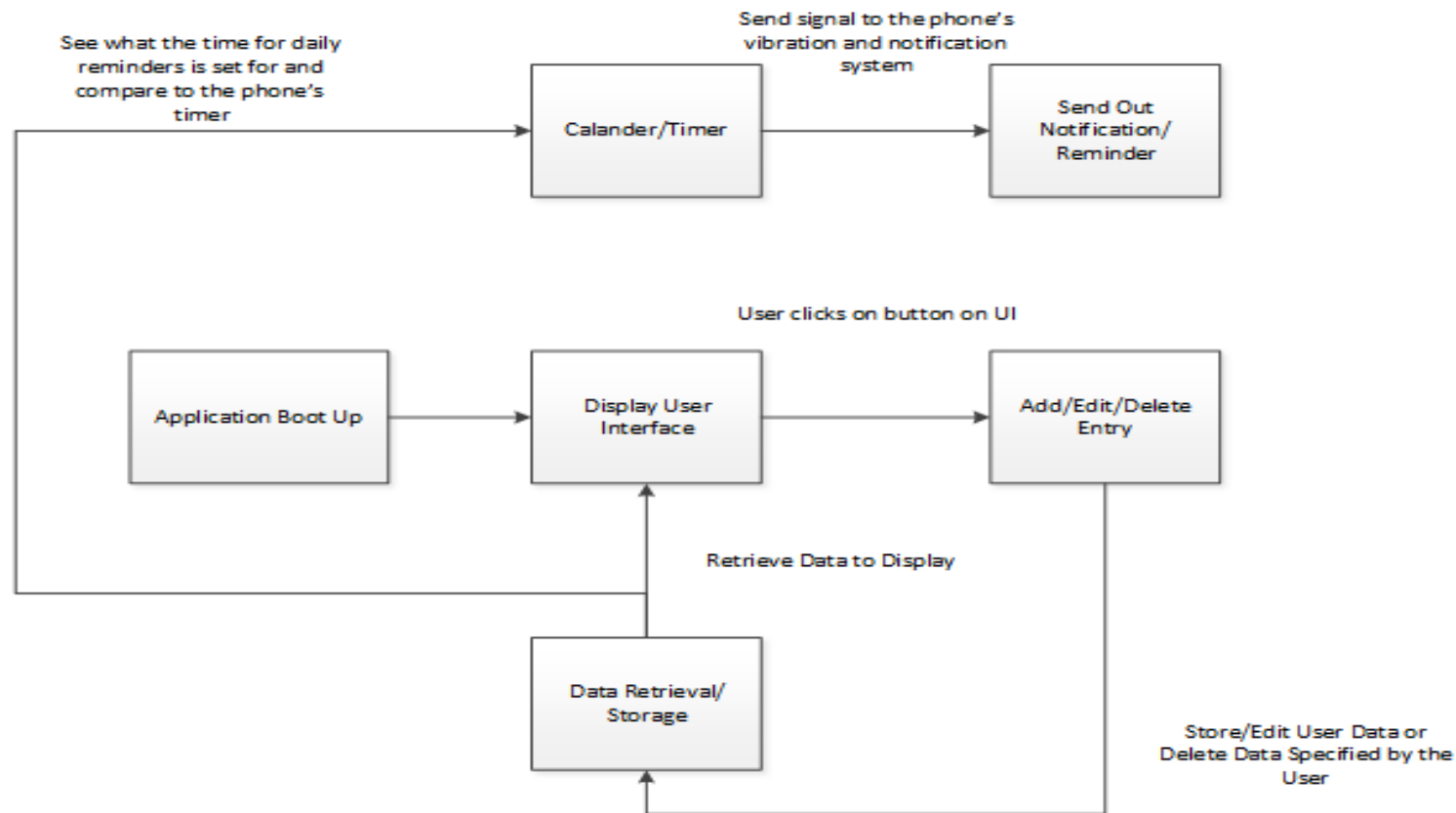
11

Anime	Manga	Books	TV	Comics	Movies	Games
<div><input type="checkbox"/> Book 1 95 pages out of 100 95%</div> <div></div> <div><input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/></div>						



System Analysis: Functional Flowchart

12



System Analysis: Classes

13

Anime	Manga
-name: String -currentEpisode: int -totalEpisodes: int -isOngoing: Boolean	-name: String -currentChapter: int -totalChapters: int -isOngoing: Boolean
+Anime(): void +getName(): String +getCurrentEpisode(): int +getTotalEpisodes(): int +getIsOngoing(): Boolean +setName(inputName: String): void +setCurrentEpisode(inputEpisode: int): void +setTotalEpisodes(inputTotal: int): void +setIsOngoing(inputFlag: Boolean): void	+Manga(): void +getName(): String +getCurrentChapter(): int +getTotalChapters(): int +getIsOngoing(): Boolean +setName(inputName: String): void +setCurrentChapter(inputChapter: int): void +setTotalChapters(inputTotal: int): void +setIsOngoing(inputFlag: Boolean): void

Comic

Book

Movie

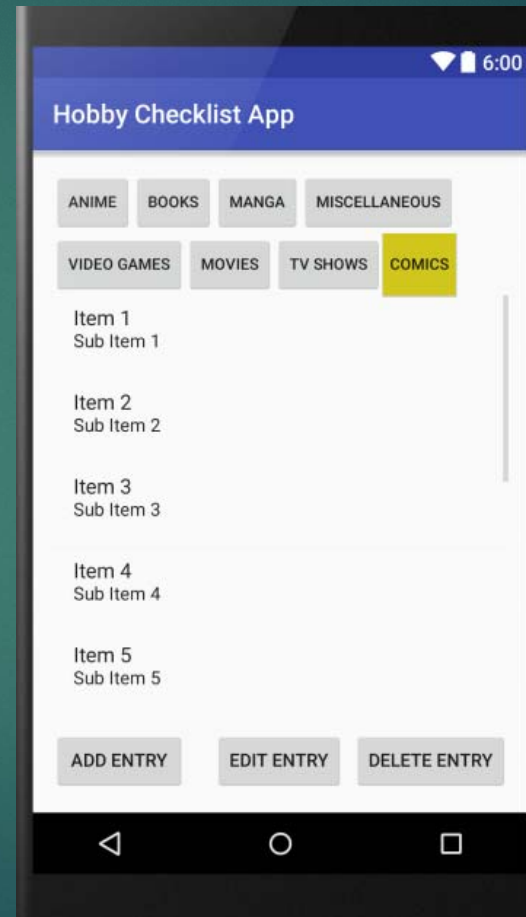
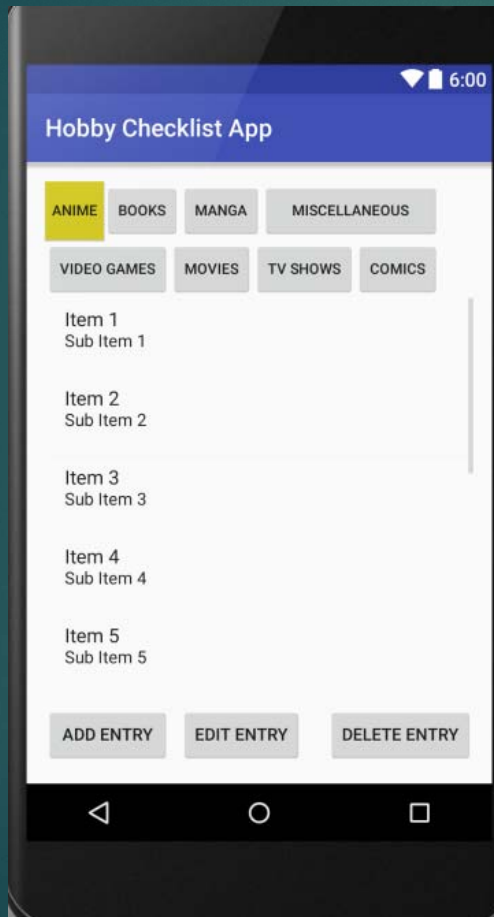
TV

Video

Misc

System Design: List Activities

14



System Design: Add/Edit Activities

15

The image displays four mobile application screens for the 'Hobby Checklist App', each showing a different 'Add/Edit' form. All screens have a blue header bar with the app name and a status bar at the top showing signal, battery, and time (6:00). The bottom of each screen features a black navigation bar with three icons: a back arrow, a circle, and a square.

- Add/Edit Anime Entry:** Includes fields for Name, Current Episode, and Total Episodes. It has a checkbox for 'Series still ongoing?', an 'ADD PICTURE' button, and 'ACCEPT'/'CANCEL' buttons.
- Add/Edit Book Entry:** Includes fields for Name, Current Page, and Total Pages. It has an 'ADD PICTURE' button and 'ACCEPT'/'CANCEL' buttons.
- Add/Edit Manga Entry:** Includes fields for Name, Current Chapter, and Total Chapters. It has a checkbox for 'Series still ongoing?', an 'ADD PICTURE' button, and 'ACCEPT'/'CANCEL' buttons.
- Add/Edit Miscellaneous Entry:** Includes fields for Name, Unit of Measure, Current Unit, and Total Units. It has an 'ADD PICTURE' button and 'ACCEPT'/'CANCEL' buttons.

System Design: Add/Edit Activities (cont)

16

The image displays four mobile application screens for the 'Hobby Checklist App', each designed for adding or editing a different type of activity. Each screen features a blue header with the app name and a status bar at the top showing a Wi-Fi icon and the time 6:00. The bottom of each screen has a black navigation bar with three icons: a back arrow, a circle, and a square.

- Add/Edit Video Game Entry:** Includes input fields for 'Name:' and 'System:', a checkbox for 'Finished?', an 'ADD PICTURE' button, and 'ACCEPT' and 'CANCEL' buttons at the bottom.
- Add/Edit Movie Entry:** Includes input fields for 'Name:' and 'Total Minutes:', a checkbox for 'Finished?', an 'ADD PICTURE' button, and 'ACCEPT' and 'CANCEL' buttons at the bottom.
- Add/Edit TV Show Entry:** Includes input fields for 'Name:', 'Season:', 'Current Episode:', and 'Total Episodes:', a checkbox for 'Series still ongoing?', an 'ADD PICTURE' button, and 'ACCEPT' and 'CANCEL' buttons at the bottom.
- Add/Edit Comic Entry:** Includes input fields for 'Name:', 'Current Volume:', and 'Total Volumes:', a checkbox for 'Series still ongoing?', an 'ADD PICTURE' button, and 'ACCEPT' and 'CANCEL' buttons at the bottom.

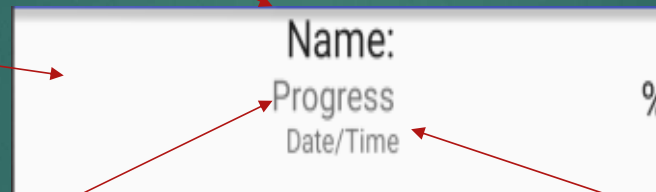
System Design: List Item Layout

17

Holds the data item's name

ImageView for displaying picture user chooses or default pic if none chosen

Calculates Percentage to a whole number and displays it



list_view_item.xml

Date and time of last update would be located here

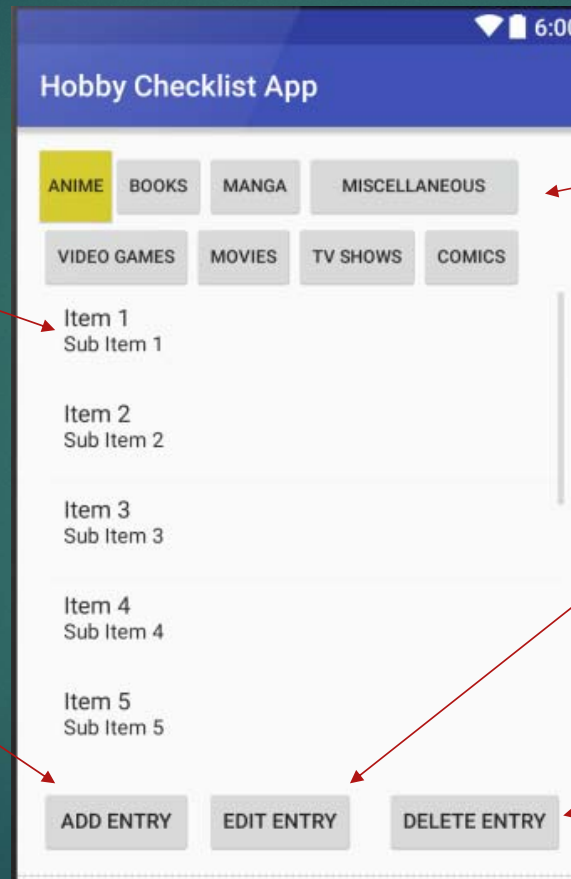
Displays the progress compared to the total number (i.e. 20 out of 50 pages)

System Design: User Interface Breakdown

18

List: Displays the entries for this class that are saved on the database

Add Button: Launches the Add/Edit Anime Activity, with blank fields



Category Buttons: Switch to the layout that will display the list of the related class/table

Edit Button: Launches the Add/Edit Anime Activity, with the fields filled in with the corresponding data for the selected entry on the list

Delete Button: Removes the selected entry on the list from database.

System Design: User Interface Breakdown

19

Ongoing Series Checkbox:
Checked if series is still
releasing new entries.

Add Picture Button:
Opens phone gallery
to choose a picture

Accept Button: Checks
the text fields and then
adds entry to database

Hobby Checklist App

6:00

Add/Edit Anime Entry

Name: _____

Current Episode: _____

Total Episodes: _____

☐ Series still ongoing?

ADD PICTURE

ACCEPT

CANCEL

Name Field: User inputs the name
of the anime.

Current Episode Field: User inputs
the episode they last left off at.

Total Episode Field: User inputs
the total number of episodes
currently out for the anime

Cancel Button: Returns to the
Anime List Activity without adding
anything to the database.

System Design: Database

20

Anime Table

Column Name	Data Type
ID	Integer
Name	String
Current Episode	Integer
Total Episodes	Integer
Is Ongoing Flag	Integer

Book Table

Column Name	Data Type
ID	Integer
Name	String
Current Page	Integer
Total Pages	Integer

Manga Table

Column Name	Data Type
ID	Integer
Name	String
Current Chapter	Integer
Total Chapters	Integer
Is Ongoing Flag	Integer

Comic Table

Column Name	Data Type
ID	Integer
Name	String
Current Volume	Integer
Total Volumes	Integer
Is Ongoing Flag	Integer

System Design: Database (cont)

21

TV Table

Column Name	Data Type
ID	Integer
Name	String
Season	Integer
Current Episode	Integer
Total Episode	Integer
Is Ongoing Flag	Integer

Video Game Table

Column Name	Data Type
ID	Integer
Name	String
System	String
Is Finished Flag	Integer

Movie Table

Column Name	Data Type
ID	Integer
Name	String
Total Minutes	Integer
Is Finished Flag	Integer

Miscellaneous Table

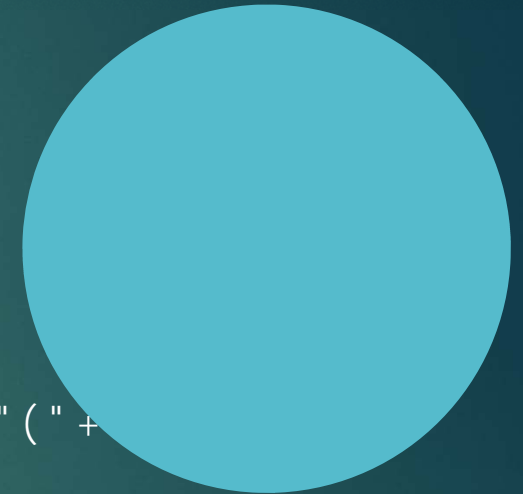
Column Name	Data Type
ID	Integer
Name	String
Unit of Measure	String
Current Unit	Integer
Total Units	Integer

System Design: Creating the Database

22

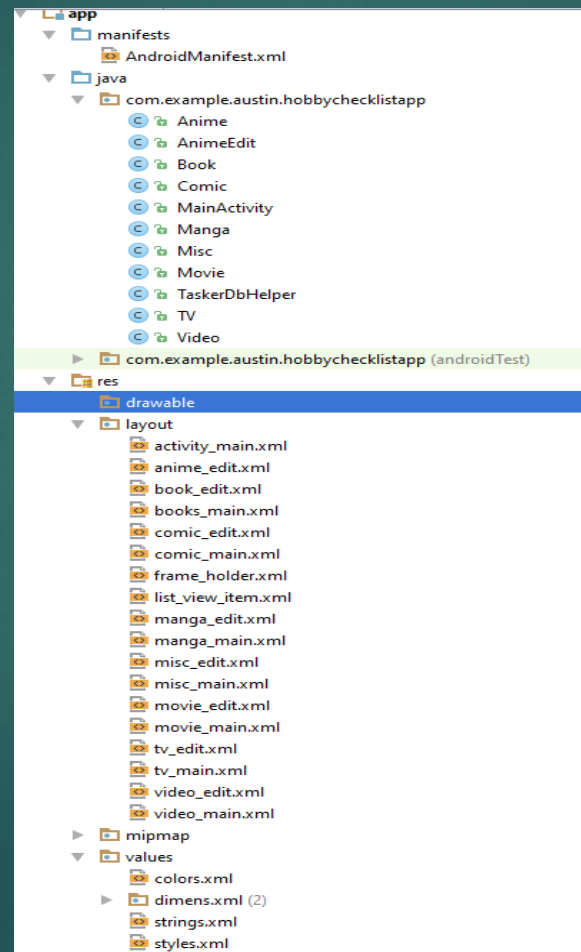
```
String sql1 = "CREATE TABLE IF NOT EXISTS " + TABLE_ANIME + " (" +  
    KEY_ID + "INTEGER PRIMARY KEY AUTOINCREMENT, " +  
    KEY_NAME + "TEXT, " +  
    KEY_CURRENT_EPISODE + "INTEGER, " +  
    KEY_TOTAL_EPISODE + "INTEGER, " +  
    KEY_IS_ONGOING + "INTEGER)";
```

```
String sql2 = "CREATE TABLE IF NOT EXISTS " + TABLE_MANGA + " (" +  
    KEY_ID + "INTEGER PRIMARY KEY AUTOINCREMENT, " +  
    KEY_NAME + "TEXT, " +  
    KEY_CURRENT_CHAPTER + "INTEGER, " +  
    KEY_TOTAL_CHAPTER + "INTEGER, " +  
    KEY_IS_ONGOING + "INTEGER)";
```



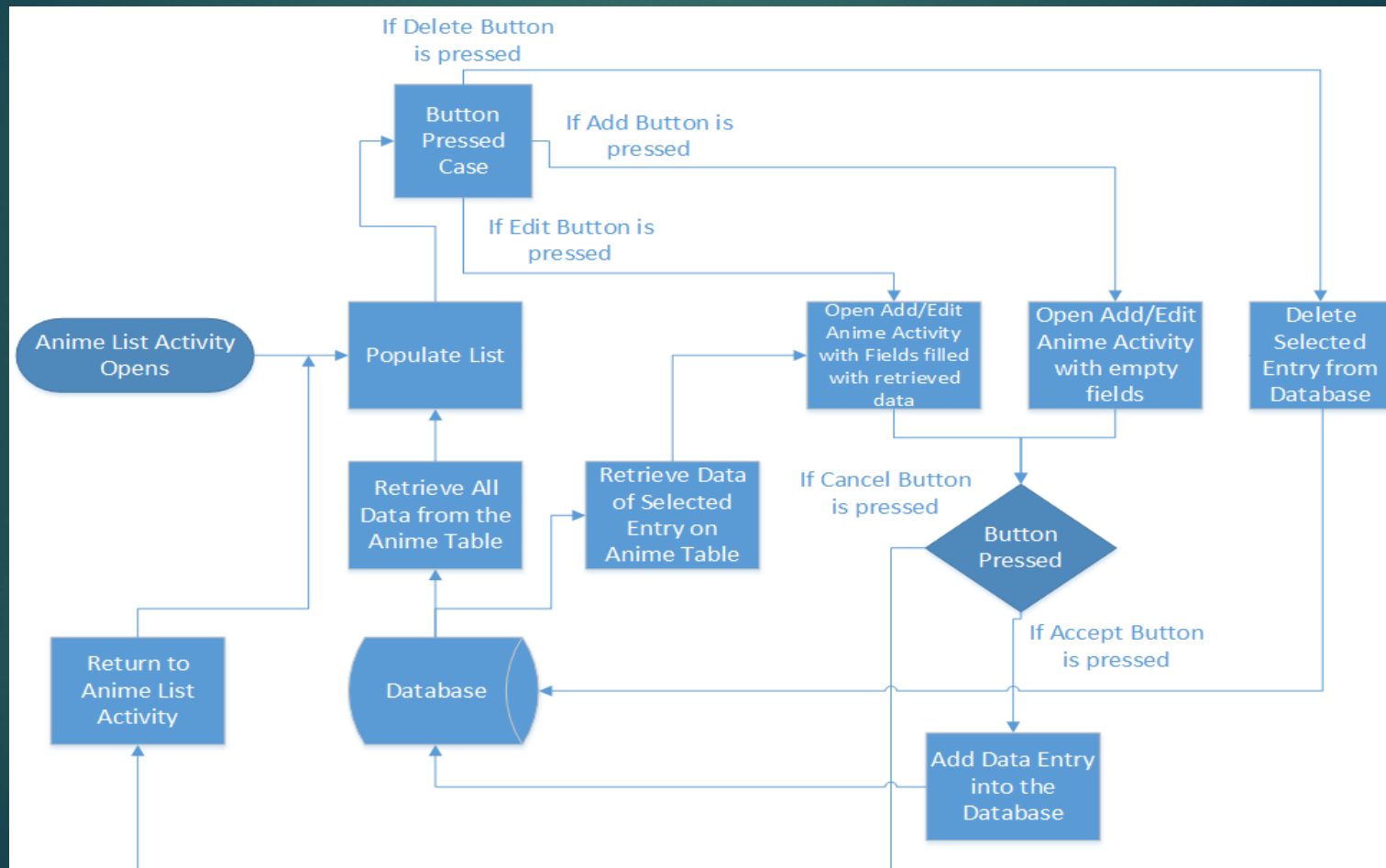
System Design: Project Structure

23



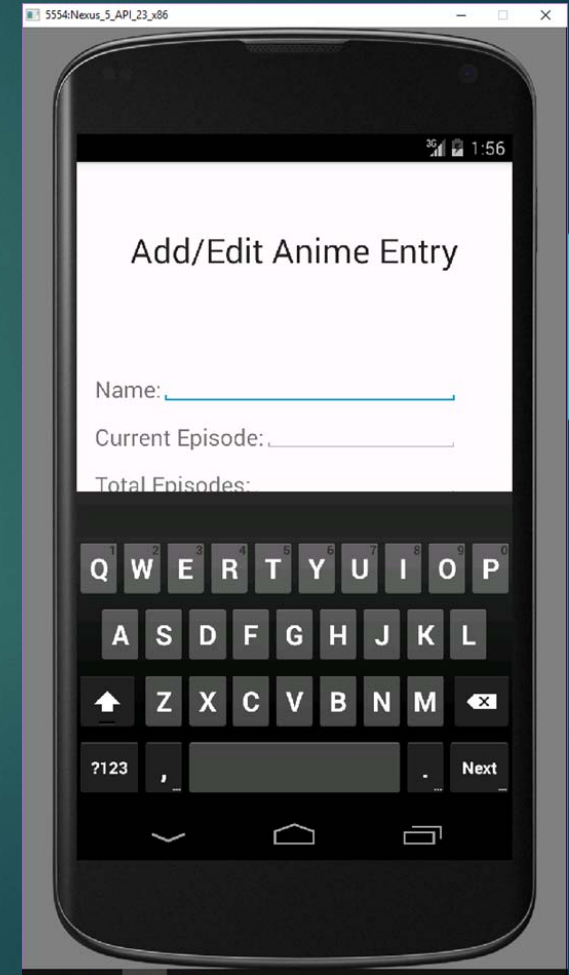
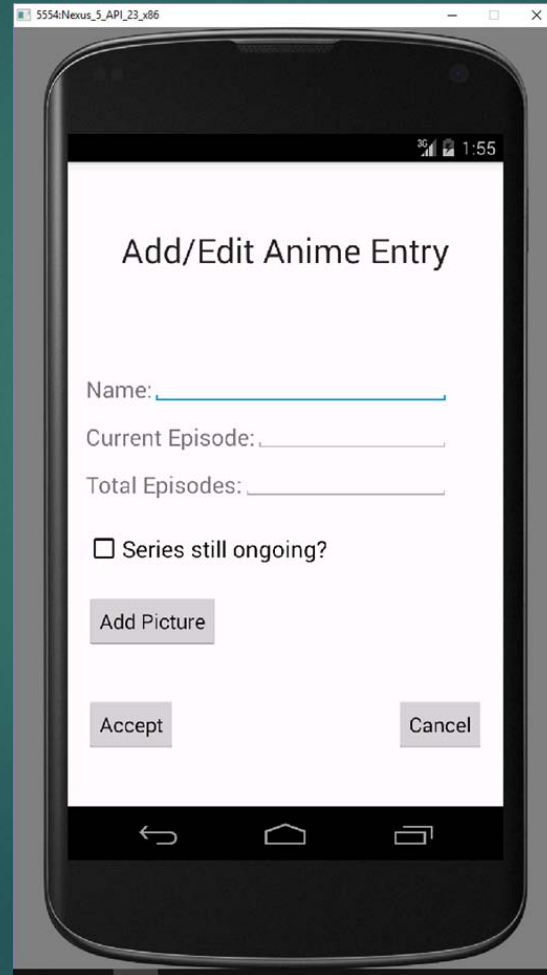
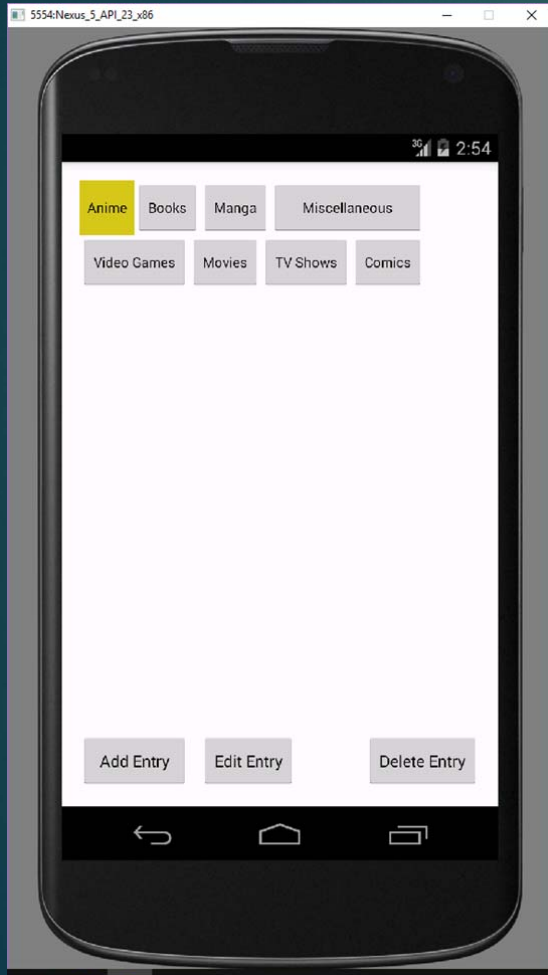
System Design: Activity Flowchart

24



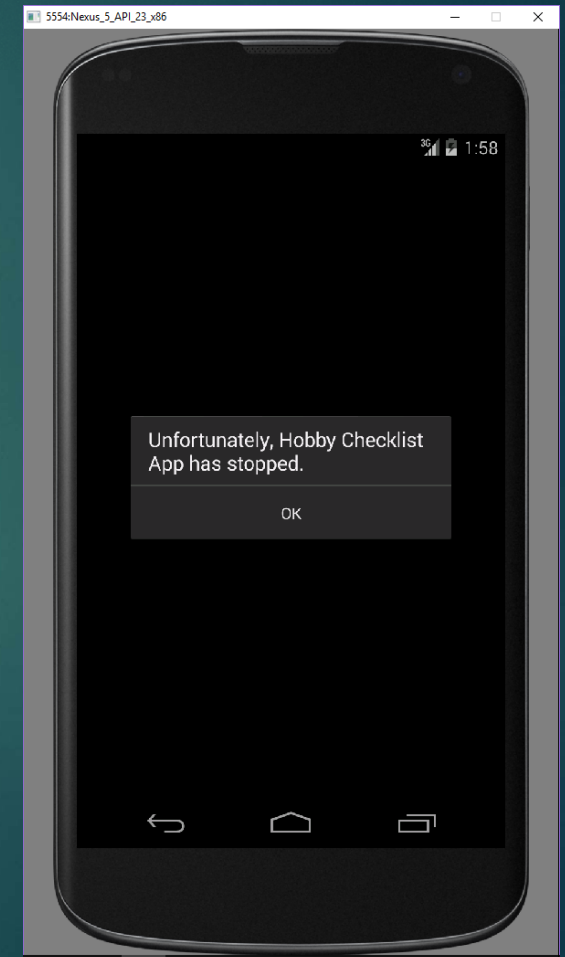
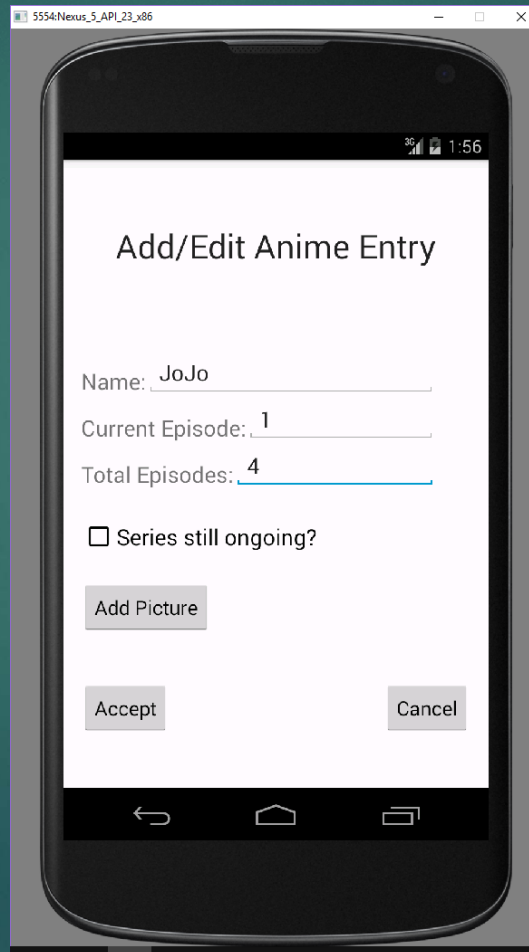
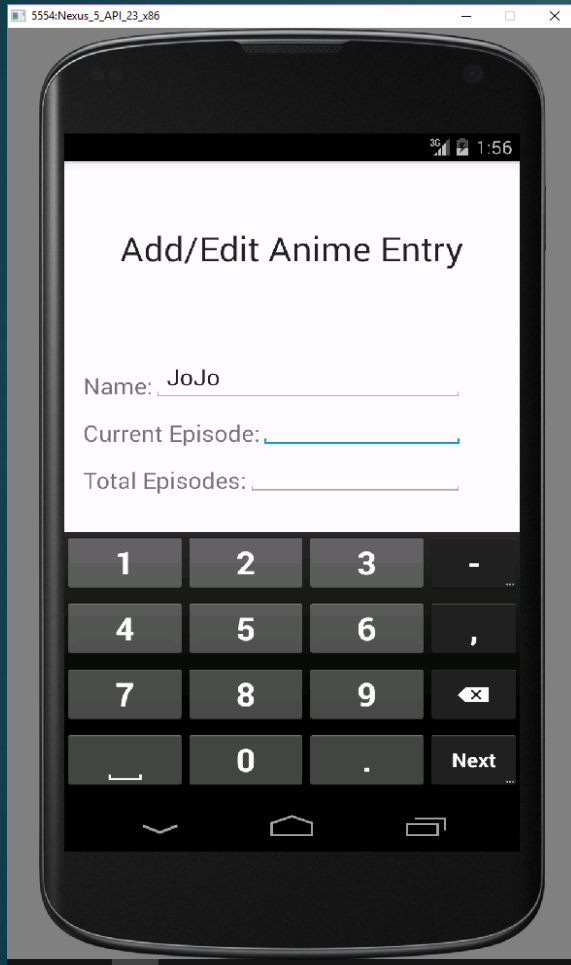
System Testing:

25



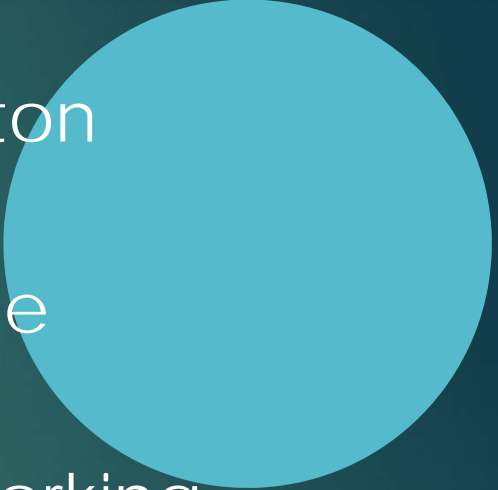
System Testing:

26



System Testing Problems

27

- ▶ App would crash when Accept button was pressed
 - ▶ Most likely caused by the code in the OnClick of the Accept button
 - ▶ So far haven't been able to get it working
- 

System Testing Problems Code

```
public void activateAnimeStorage(View view) {  
    if (checkEditText() == false && checkValues() == false){  
        setAnimeValues();  
        dbHelper.addAnimeItem(mAnime);  
  
        Intent launchMain = new Intent(this, MainActivity.class);  
  
        startActivity(launchMain);  
    }  
}
```


Validations

29

Requirement Data		
ID	Requirement Type	Requirement (Shall or Should statements)
1	Functional	The application shall provide a user interface
2	Operational	The application shall keep track of the user's progress on a set of leisure activities
3	Operational	The application shall send out daily notifications using the phone's vibration function that will leave a text message to the user.
4	Functional	The application shall store data on the phone's storage
5	Functional	The project shall organize activities into categories
6	Performance	The categories shall be Anime, Manga, Comics, TV, Movies, Books, Video Games, and Miscellaneous
7	Functional	The project shall use the phone's vibration function for notifications.
8	Functional	The application shall use the phone's timer to know when to send daily notification
9	Performance	The application shall display progress completed up to the nearest % value based on the completion of the activity (ex. 95 pages out of 100 pages would be 95%)
10	Performance	The application shall send out the daily notification at noon local time +/- 5 minutes
11	Environmental	The application shall work on Android OS 4.4.2 on Samsung S3
12	Environmental	The application should work on Android OS 4.4.3 and later
13	Physical	The project shall reserve 300 MB for storage use
14	Physical	The project shall not use more than 500 MB of CPU
15	Physical	The application shall not be larger than 700 MB
16	Functional	The daily notification time should be user-specified
17	Environmental	The application shall be developed under the Android Studio programming environment
18	Functional	The user shall not be able to edit the category names
19	Functional	The application shall retrieve data off the phone's storage

Schedule Management

30

Task Name	Duration	Start	Finish
Phase II Start	0 days	Mon 1/18/16	Mon 1/18/16
System Design Complete	0 days	Tue 2/9/16	Tue 2/9/16
Detailed Design Complete	0 days	Mon 3/14/16	Mon 3/14/16
Functional Prototype Complete	0 days	Fri 4/1/16	Fri 4/1/16
Verification Testing Complete	0 days	Thu 4/21/16	Thu 4/21/16
Final Report Submitted	0 days	Mon 8/25/14	Mon 8/25/14
Presentation Delivered	0 days	Mon 8/25/14	Mon 8/25/14
End of Semester	0 days	Mon 5/9/16	Mon 5/9/16
Prototype Development	69 days	Mon 1/18/16	Thu 4/21/16
System Design Phase	17 days	Mon 1/18/16	Tue 2/9/16
Practice With Development Software	14 days	Mon 1/18/16	Thu 2/4/16
Create Diagrams	7 days	Mon 1/18/16	Tue 1/26/16
Update System Design	3 days	Fri 2/5/16	Tue 2/9/16
Final System Design Complete	0 days	Tue 2/9/16	Tue 2/9/16
Detailed Design Phase	24 days	Wed 2/10/16	Mon 3/14/16
Software Design	24 days	Wed 2/10/16	Mon 3/14/16
Create The User Interface	3 days	Wed 2/10/16	Fri 2/12/16
Code the App	21 days	Mon 2/15/16	Mon 3/14/16
System Integration Phase	14 days	Tue 3/15/16	Fri 4/1/16
Code the Functions for the User Interface	7 days	Tue 3/15/16	Wed 3/23/16
Code the Daily Notifications Function	7 days	Thu 3/24/16	Fri 4/1/16
System Verification Phase	14 days	Mon 4/4/16	Thu 4/21/16
Troubleshoot the App	7 days	Mon 4/4/16	Tue 4/12/16
Make changes as needed to fix any problems	7 days	Wed 4/13/16	Thu 4/21/16
Verification Testing Complete	0 days	Thu 4/21/16	Thu 4/21/16



Schedule Management

31

- ▶ Stayed close to the planned schedule in the first 2 months.
- ▶ When the Coding phase started, that's when the schedule got set back.
- ▶ Originally planned to have coding done mid-March

Risk Management

32

ID	Entry Date	Type		Risk Description: 'IF statement'		Consequence of Risk: 'THEN statement'	Status	Likelihood (1-5)	Severity (1-5)	Score	Rank*	Response	Description of Response
1	9-Nov-15	Cost	IF	App development software is not available in a free format	THEN	A software will be needed to be purchased	Open	2	2	4	Low	Accept	
2	9-Nov-15	Schedule	IF	The software is incompatible with the phone being used	THEN	Time would be spent looking an alternative software and put schedule at risk	Open	2	4	8	Medium	Avoid	Ensure that the software works on the phone before getting too in depth
3	9-Nov-15	Technical	IF	I am unable to fully comprehend the language	THEN	The prototype could be unable to perform as required	Open	2	5	10	Medium	Avoid	Seek help and allocate time to learning the language
4	9-Nov-15	Cost	IF	The phone I am currently using is unable to connect to the computer	THEN	A new phone will be needed to continue the project	Open	2	2	4	Low	Accept	
5	9-Nov-15	Schedule	IF	The computer at home being used to develop the app crashes	THEN	App would need to be rewritten putting the project behind schedule	Open	2	5	10	Medium	Transfer	Use flash drive as back up drive to ensure data is not lost
6	9-Nov-15	Technical	IF	The app does not successfully manage the data	THEN	The prototype would not meet one or more system requirements	Open	3	5	15	High	Avoid	Ensure that the app retrieves and stores data properly so data on user interface is accurate
7	9-Nov-15	Technical	IF	The app does not send out the daily notifications	THEN	The prototype does not meet one or more system requirements	Open	2	4	8	Medium	Avoid	Ensure that the app performs the function properly
8	27-Nov-15	Cost	IF	Project requires more labor hours than predicted	THEN	More labor hours will be needed in order to not put schedule at risk	Open	2	4	8	Medium	Mitigate	Use more weekend time to accommodate need for more labor hours

Risk Management

- ▶ Saved program data onto external flash drive for back up incase something were to happen to my computer
- ▶ No cost risks occurred as the Android Studio software worked for me
- ▶ Could have handled the risk involving my inexperience with Android App Development better

Lessons Learned

- ▶ I got a better understanding of what my strengths and weaknesses are so that I can focus on the areas that need improvement.
- ▶ I realized that I need to sometimes ask for help when I need it and not try to do everything by myself when I struggle.
- ▶ Acquired a greater understanding of the processes of the Project Lifecycle and everything that goes into managing a project.
- ▶ Learned what the failure of a project is like the hard way.

► Questions?

