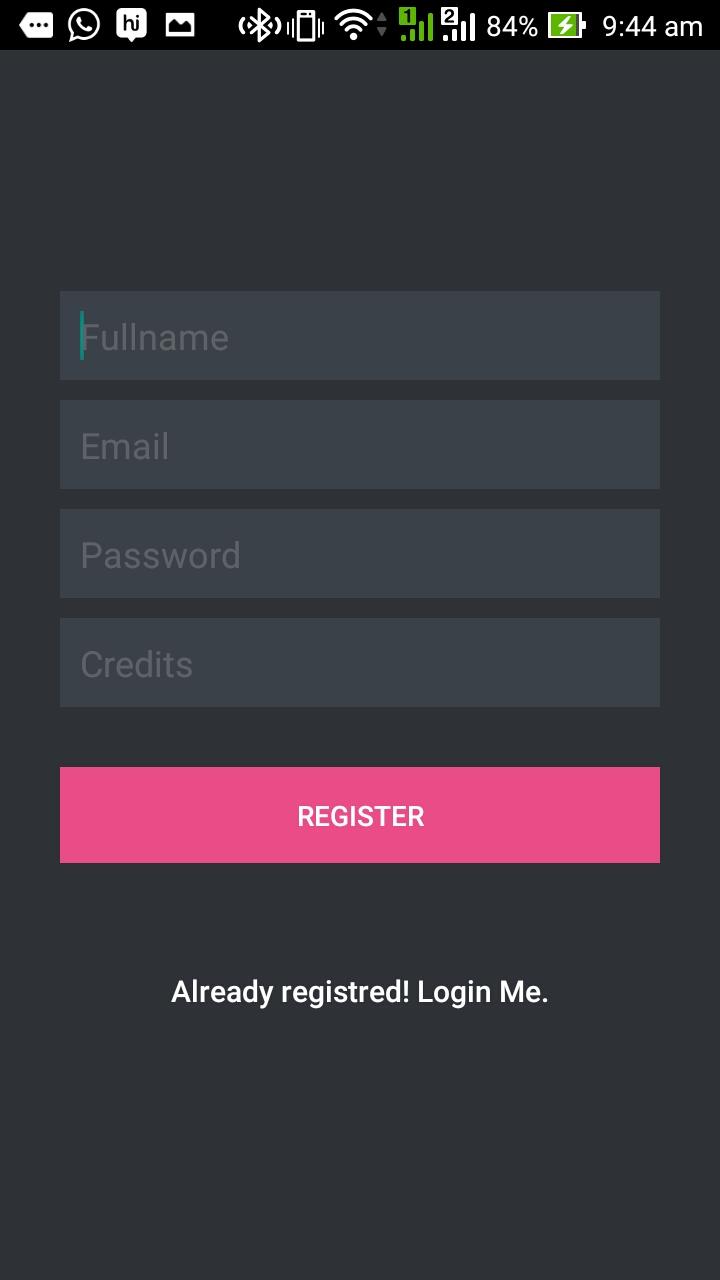
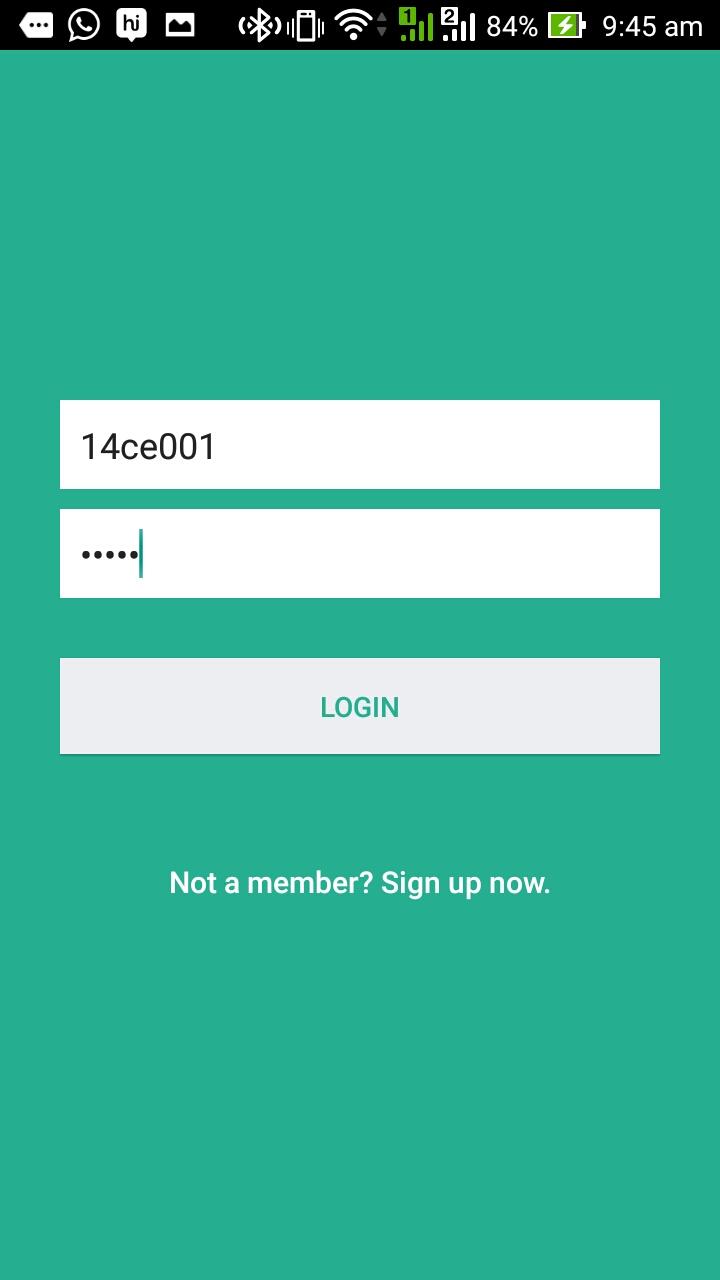
**A**

**Project Report**

**On**

**Cash Free**

**  **

**Prepared by**

14CE098 Mikin Patel

14CE107 Varsh Patel

**Under the guidance of**

Prof. Hina Vaghashia

**Submitted to**

Charotar University of Science & Technology

Degree of Bachelor of Technology

in Computer Engineering

CE222.01 -Software Group Project

Of 4th Semester of B.Tech

**Submitted at**



**U. & P. U PATEL DEPARTMENT OF COMPUTER ENGINEERING**

**Faculty of Technology & Engineering, CHARUSAT**

**Chandubhai S. Patel Institute of Technology**

**At: Changa, Dist: Anand – 388421**

**January-April 2016**





This is to certify that the report entitled “Typing Tutor” is a bonafied work carried out by **Mr Mikin Patel(14CE098)** and **Mr Varsh Patel(14CE107)** under the guidance and supervision of Prof. Hina Vaghashiafor the subject **Software Group Project (CE222.01)** of 4th Semester of Bachelor of Technology in **Computer Engineering** at Faculty of Technology & Engineering (C.S.P.I.T.) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

|  |  |
| --- | --- |
| Under the supervision of,  Prof. Hina Vaghashia  Assistant Professor  U. & P. U Patel Dept. of Computer Engg.  C.S.P.I.T., CHARUSAT-Changa. |  |
| Dr. (Prof.) Amit Ganatra  Dean,  Faculty of Technology & Engineering  Head, U. & P. U Patel Department of Computer Engineering  C.S.P.I.T., CHARUSAT- Changa, Gujarat. | |

**Chandubhai S Patel Institute of Technology (C.S.P.I.T.)**

**Faculty of Technology & Engineering, CHARUSAT**

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

**Abstract**

This project is a student utility android application developed specifically for CHARUSAT, Changa education campus currently. The students are facilitated to pay money to canteen, stationary and library in the form of credits that they have earned from admin by giving its equivalent cash. There is registration page from where a student can get registered to the app and later the same credentials can be used to login to the application. Inside application there are main 5 modules containing canteen pay, stationery pay, library fine, options and wear connect respectively. Further, options module has functionality for admin to refill credits followed by show credits module that shows the whole transaction per student logged in. At the end, when a student logout then the corresponding credits and details get updated to the database.

**Acknowledgement**

I am thankful to my internal guide Prof. Hina Vaghashia from Charusat for her constant and valuable guidance at each and every stage of the project. I would like to thank all my family members and friends for their support. Also extending my thanks to the rising technology in android that made me to select this project with great passion.

**Table of contents**

Abstract

Acknowledgement

Chapter1 Introduction

1.1. Purpose

1.2. Scope

1.3. Definitions, acronyms & abbreviations

1.4. References

1.5. Overview

Chapter 2 Overall description

2.1. Product perspective

2.1.1. System interfaces

2.1.2. User interfaces

2.1.3. Hardware interfaces

2.1.4. Software interfaces

2.1.5. Communications interfaces

2.1.6. Memory constraints

2.1.7. Operations

2.1.8. Site adaptation requirements

2.2. Product functions

2.3. User characteristics

2.4. Constraints

2.5. Assumptions and dependencies

Chapter 3 System Analysis

3.1 Requirements of New System

3.1.1 Functional Requirements

3.1.2 Non Functional Requirements

3.2 Feasibility Study

3.2.1 Does the system contribute to the overall objectives of the

organization?

3.2.2 Can the system be implemented using the current technology and

within the given cost and schedule constraints?

3.2.3 Can the system be integrated with other system which are already in

place?

3.3 System Activity (Use case and/or scenario diagram)

3.4 Class Diagram

3.5 Sequence Diagram

3.6 Functions of system

3.7 Context Diagram

3.8 Data Flow Diagram (0 and 1 level or higher)

3.9 Data Modeling

3.9.1 ER Diagram

* 1. Selection of Hardware and Software and Justification

Chapter 4 Project estimation

* 1. Estimation technique used
  2. Effort resource, cost, project duration estimation

Chapter 5 Schedule in which

* 1. breakdown structure
  2. Task network representation
  3. Gantt chart representation
  4. Pert chart representation

Chapter 6 Project resources

* 1. People
  2. Hardware & Software
  3. Special resources

Chapter 7 Project tracking & control plan

Chapter 8 System Design

8.1 System Application Design

* + 1. Method Pseudo code
    2. Samples Of Forms, Reports and Interface

Chapter 9 Implementation Planning

a. Implementation Environment (Single vs Multiuser, GUI vs Non GUI)

* 1. Program/Modules Specification(refer appendix C)
  2. Security Features
  3. Coding Standards

Chapter 10 Testing

1. Testing Plan
2. Testing Strategy
3. Testing Methods
4. Test Suites Design
5. Test Cases
   * 1. Purpose
     2. Required Input
     3. Expected Result

Chapter 11 Limitation and Future Enhancement

Chapter 12 Conclusion and Discussion

1. Self Analysis of Project Viabilities
2. Problem Encountered and Possible Solutions
3. Summary of Project work

**CHAPTER 1 INTRODUCTION:**

**1.1. PURPOSE:**

The main purpose of the specification is to guide the developer about the application and the development structure, admin and student services, requirements and the input and output feature.

**1.2. SCOPE:**

The Cash Free application has the ability to provide an cah free transaction utility to college going students so that they don’t require to carry risk of money in their wallet. They can directly use their credits available in their account in canteen, stationary and admin.

BENEFITS:

* Saves time.
* Reduces risk.
* No need to argue on not having change.

OBJECTIVES:

* 0% risk of cash in wallet.
* Efficient completion of the transaction in respective areas in university campus.

**1.3. DEFINITIONS, ACRONYMS AND ABBREVIATION:**

* AW – Android Wear
* AD – Android Device
* AA – Android Application
* CSA – Canteen, Stationary and Admin
* Trac. – Transaction
* Crd. – Credits

**1.4. REFERENCES:**

The books and materials referred during the pre-development stages of the project include:

* <http://developer.android.com>
* <http://www.wikipedia.org>
* Software Engineering by Rajib Mall
* http://www.w3schools.com

**1.5. OVERVIEW:**

Section I gives a brief introduction about the document and the objectives of the software. The overall description such as system interface, software interface, hardware interface, communication interface are discussed in section II and the software product featured section III.

**CHAPTER 2 OVERALL DESCRIPTION:**

**2.1. PRODUCT PERSPECTIVE:**

The product works on any AD and AW connected to it. It can have a unique login per AD that is linked to the server which contains MYSQL and php present and can fully complete the transaction ones initiated.

**2.1.1. System Interfaces:**

The AD should be able to share the Crd. data available in the account among the AW connected.

The AD should be able to connect to the server to fetch all the databases present via php pages up on the server.

**2.1.2: User Interfaces:**

The screen format in the application is in the menu type which contains sub-menus in frame format on the main screen, which is used for the interaction with the user. This menu type screen format allows the user to choose his options (i.e. CSA). This type of format will make the user to operate easily with more accuracy.

**2.1.3: Hardware interfaces:**

* Mobile phone.
* Android Wear (optional).

**2.1.4: Software Interfaces:**

|  |  |  |
| --- | --- | --- |
| NAME | VERSION | SOURCE |
| Android OS | 2.3 or higher | Google Android |
| Android Wear OS | 4.1.1 or higher | Google Android |
| Wamp Server | 3.0.0 or higher | Wamp server |

**2.1.5: Communication Interface:**

This application can run on an active Wireless Internet Facility.

**2.1.6: Memory Constraints:**

The system would require a free space of 50 MB in AD with RAM greater than 512 MB and 4 MB space in AW.

**2.1.7: Operations:**

The student will be in need to pay cash to admin on the end of every month to get Crd. in return equivalent of cash payed, later that Crd. Can be used by students to pay at CSA for buying the products. The students can directly pay Crd. on to respective CSA account and can get his product in seconds. The application will handle the overall Trac. After every month end.

**2.1.8: Site Adaptation Requirements:**

The application requires no special modifications to adapt to particular installations except for the files that are required to be re-created.

**2.2. PRODUCT FUNCTIONS:**

The major function includes:

* Providing cash to the administrator at the end of every month to avail Crd.
* Using the Crd. At CSA.
* Each student has their personal account in their AD.
* Students can see their remaining Crd. On their AW connected to mobile.
* Updating Trac. After evry month to CSA respectively.

**2.3 USER CHARACTERISTICS:**

It is sufficient that the users have a basic knowledge needed for operating the application product. There is no need for any experience or technical knowledge.

**2.4. CONSTRAINTS**

Not applicable.

**2.5. ASSUMPTIONS AND DEPENDENCIES**

Every students has an AD with their account created at the time of admission to the university, though it is not compulsory to add credits on to the account. This application is limited to pay cash via Crd. In CSA only and not beyond that. We assume that at the end of every month whatever the transactions are left for CSA will be done by administration of the college itself.

**CHAPTER 3 SYSTEM ANALYSIS:**

**3.1 REQUIREMENTS OF NEW SYSTEM**

* + 1. **Functional Requirements**

R.1 Register-

Description: User can register a new username.

Input: Enter Name, StudentID, Password, Initial credits and click on register.

Output: Redirected to Login.

R.2 Login-

Description: User can login with registered StudentID and password.

Input: Enter StudentID and Password then click on login.

Output: Redirected to Homepage.

R.3 Homepage

Description: Student can choose category of payment i.e. CSA also connect to wear notification and options page.

Input: Select a specific category.

Output: Redirected to selected category.

R.4 Canteen, Stationery, Library

Description: Student can pay credits to canteen, stationery and liberary respectively.

Input: Enter amount to pay and click pay.

Output: Deducts credits from student credits and respective category database is updated with credits.

R.5 Options

Description: Student can get transaction details and admin panel is accessible form here.

Input: Choose required option.

Output: Redirected to input option.

R.6 Wear

Description: Student can get a notification on wear connected.

Input: No input required.

Output: Gets notification in wear connected to the device.

**3.1.2 Non Functional Requirements**

* Usability- Software is easy to use.
* Reliable- There is less probability of failures.
* Portability- It can be used in Android Wear.
* Fault tolerance- In case of failure, the system can restart quickly.
* Extensibility- New features can be added to it in the future.

**3.2 FEASIBILITY STUDY**

**3.2.1 Does the system contribute to the overall objectives of the**

**organization?**

Yes, the system developed contributes to all objects.

**3.2.2 Can the system be implemented using the current technology and**

**within the given cost and schedule constraints?**

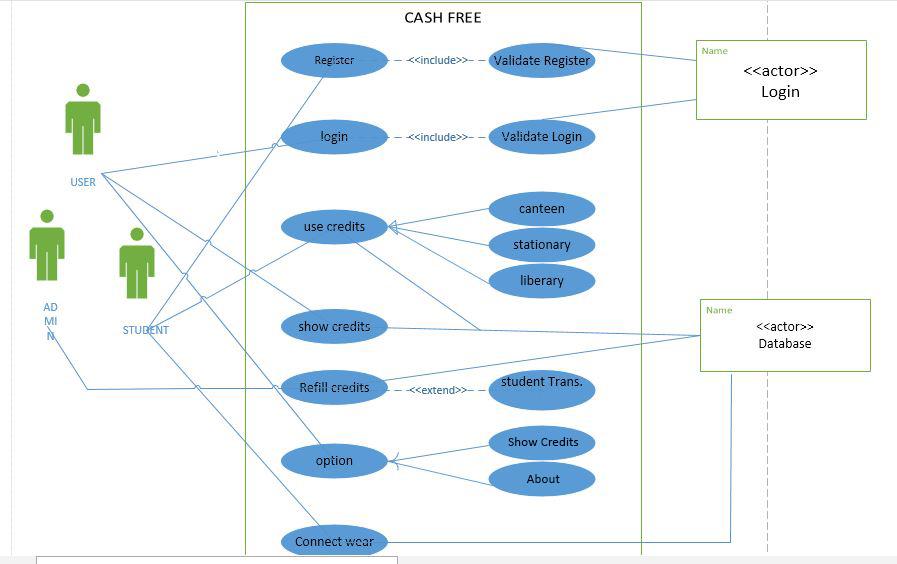
Yes, the system can be implemented using java and mysql and with the schedule constraints

**3.2.3 Can the system be integrated with other system which are already in**

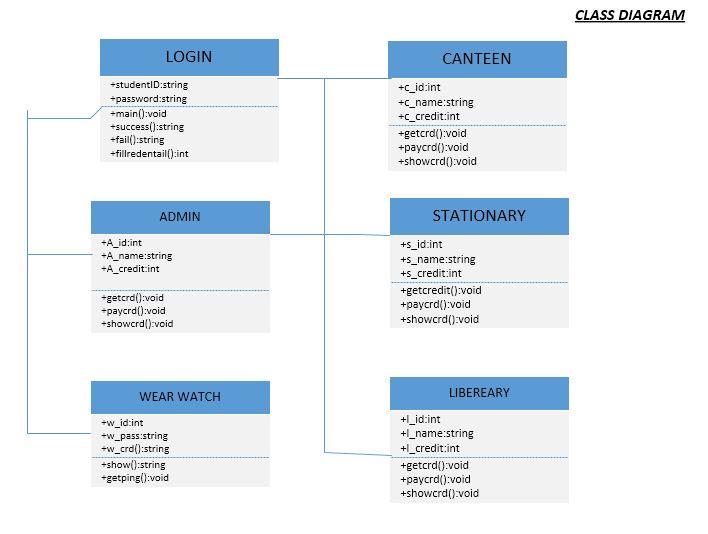
**place?**

Yes, the system can be integrated with other systems.

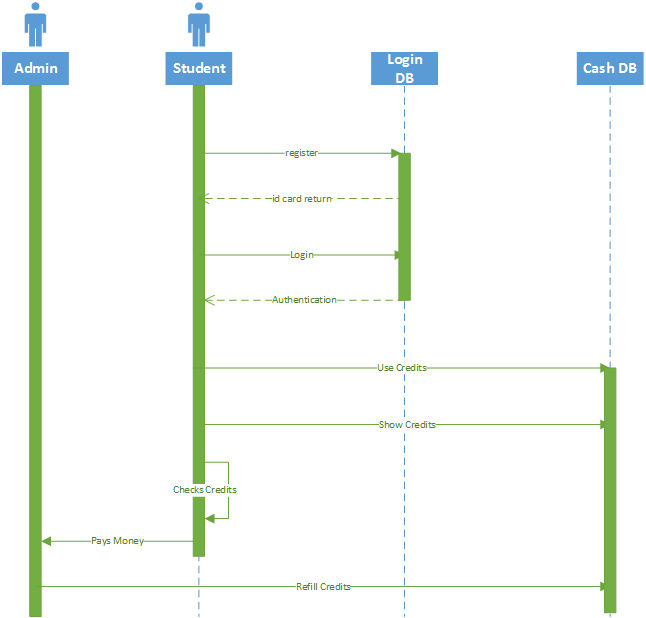
**3.3 SYSTEM ACTIVITY**



**3.4 CLASS DIAGRAM**



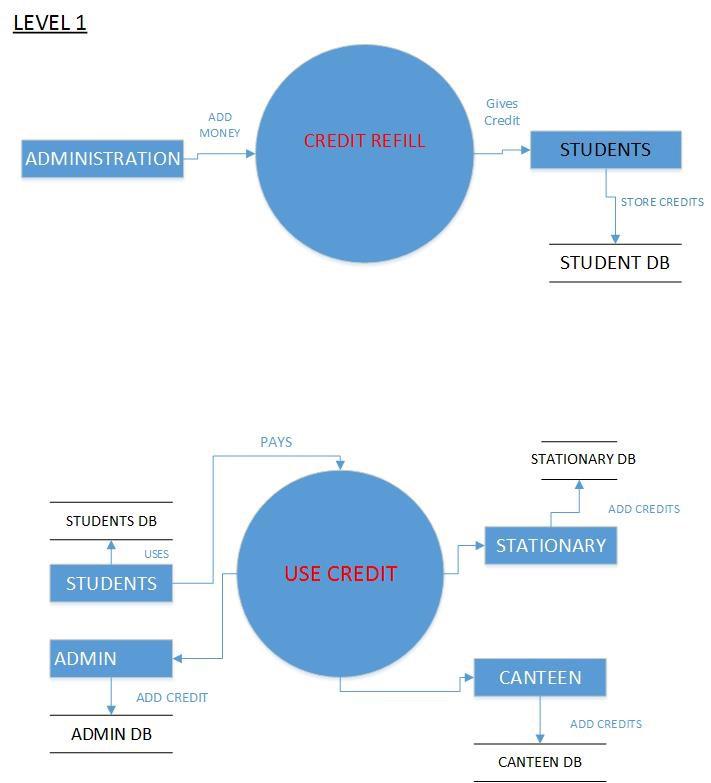
**3.5 SEQUENCE DIAGRAM**



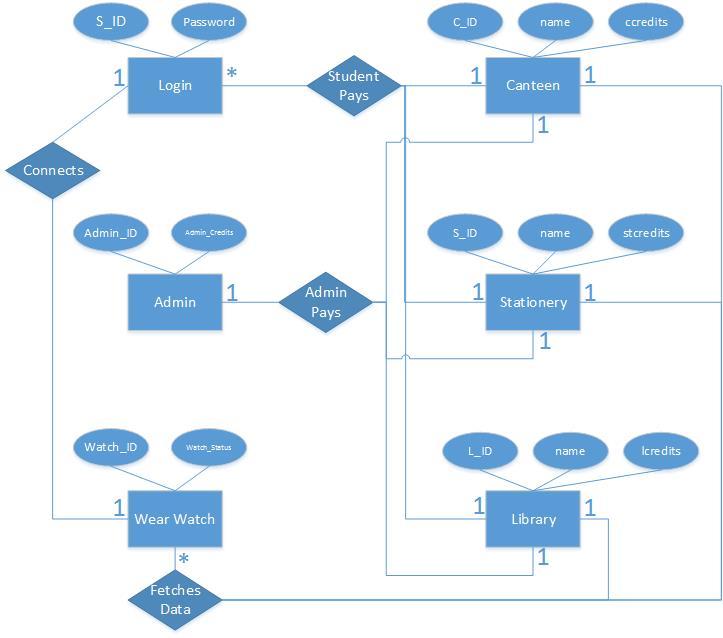
* 1. **FUNCTIONS OF SYSTEM**
* Create a username- User can create a new username.
* Login- User can login with existing username.
* Practice typing- User can practice typing in any of the 3 levels: beginner, intermediate or expert.
* Test- User can give a test to see how he/she has progressed.
* Progress report- User can view their progress report to see how they have progressed.
  1. **CONTEXT DIAGRAM**



**3.8 DATA FLOW DIAGRAM**



* 1. **DATA MODELING**
     1. **ER Diagram**



**3.10 SELECTION OF HARDWARE AND SOFTWARE AND JUSTIFICATION**

We have selected android studio, AD, AW, php support, and MYSQL to make this project. Using these software we can implement the project and give all the functionalities.

**CHAPTER 4 PROJECT ESTIMATION:**

**4.1 ESTIMATION TECHNIQUE USED:**

The project size is a measure of the problem complexity in terms of the effort and time required to develop the product. Currently, two metrics are popularly being used to estimate size: Line of code (LOC) and function point (FP).

* 1. **EFFORT RESOURCE, PROJECT DURATION ESTIMATION:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Parameters** | **Count** |  | **Simple** | **Average** | **Complex** |  | **Total** |
| No. of user Input | 4 | X | 3 | 4 | 6 | = | 12 |
| No. of user Output | 4 | X | 4 | 5 | 7 | = | 16 |
| No. of Inquires | 2 | X | 3 | 4 | 6 | = | 6 |
| No. of Files | 4 | X | 7 | 10 | 15 | = | 28 |
| external Interface | 1 | X | 5 | 7 | 10 | = | 5 |

**Complexity Weight Factor:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Factors** | **Weights** |
| 1. | Does the system require reliable backup and recovery? | **2** |
| 2. | Are data communication required? | **4** |
| 3. | Are there distributed processing functions? | **4** |
| 4. | Is performance critical? | **3** |
| 5. | Will the system run in an existing, heavily utilized operational environment? | **3** |
| 6. | Does the system require online data entry? | **5** |
| 7. | Does the on-line data entry require the input transactions to be built over multiple screens or operation? | **5** |
| 8. | Are the master file updated on-line | **4** |
| 9. | Are the inputs, outputs, files, or inquiries complex? | **2** |
| 10. | Is the internal processing complex? | **2** |
| 11. | Is the code designed to be reusable? | **4** |
| 12. | Are conversion and installation included in the design? | **2** |
| 13. | Is the system designed for multiple installations in different organizations? | **5** |
| 14. | Is the application designed to facilitate change and ease of use by the user? | **5** |

|  |  |
| --- | --- |
| **Weight** | **Degree Of Influence** |
| 0 | No Influence |
| 1 | Incidental |
| 2 | Moderate |
| 3 | Average |
| 4 | Significant |
| 5 | Essential |

**FP Count:**

FP = count total\*[0.65+0.01 \*∑(Di)]

FP = 67\* [0.65+0.01\*50]

FP = 77.05

**Function Point is: 77.05**

Line of code (LOC) = FP\*30 = 77.05\*53 = 2311.50

KLOC = 2.31150

**Software Project Type**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **ab** | **bb** | **cb** | **db** |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| **Semi-detached** | **3.0** | **1.12** | **2.5** | **0.35** |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Effort = ab\*(KLOC) ^bb

= 3.0\*(2.31150) ^1.12

= 7.67 PM

Tdev = cb \* (Effort) ^db

= 2.5\*(7.67) ^0.35

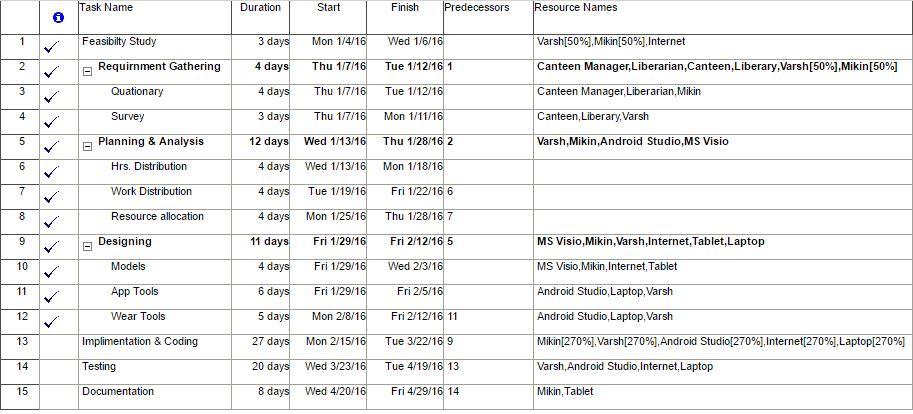
= 5.10 Months

**CHAPTER 5 SCHEDULE IN WHICH**

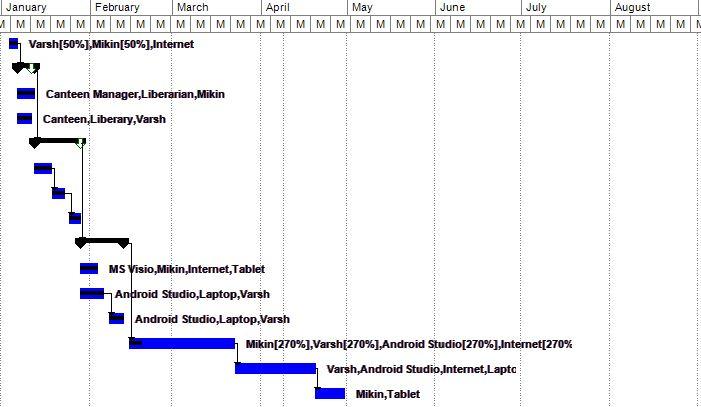
* 1. **BREAKDOWN STRUCTURE**



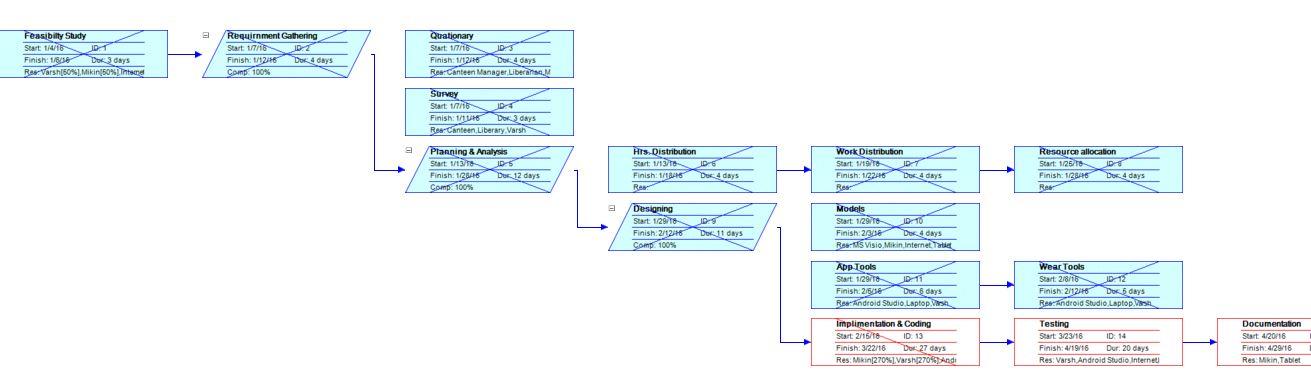
**5.2 TASK NETWORK REPRESENTATION**



* 1. **GANTT CHART REPRESENTATION**



* 1. **PERT CHART REPRESENTATION**



**CHAPTER 6: PROJECT RESOURCES**

* 1. **PEOPLE**

|  |  |
| --- | --- |
| **Developer** | **Roles and Responsibilities** |
| 14CE098 | Requirement gathering, analysis, coding |
| 14CE107 | Designing, coding, testing |

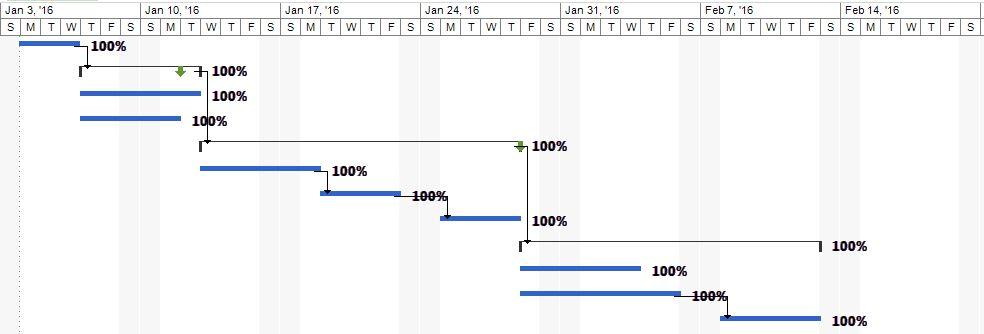
User: Any user having basic knowledge of computer can use this software.

Developer: Developer with knowledge of Android app development and knowledge of working with servers and php with MYSQL can use this app.

* 1. **HARDWARE AND SOFTWARE**

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| AD Specifications:  RAM:4 GB  Hard Disk: 16 GB  Android Version: 5.0.1  AW Specifications:  RAM:1 GB  Hard Disk: 20 MB  Android Wear Version: 4.2.1 | Android Studio 1.5.1  MySQL 5.7.9  WampServer 3.0.0 |

**CHAPTER 7: PROJECT TRACKING AND CONTROL PLAN**



**CHAPTER 8: SYSTEM DESIGN**

* 1. **SYSTEM APPLICATION DESIGN**
     1. **Method Pseudo code**

One of the function of the project:

Canteen Module:

**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_canteen***);  
 **tv**=(TextView) findViewById(R.id.***textView2***);  
 **editText** = (EditText) findViewById(R.id.***editText***);  
 **button** = (Button) findViewById(R.id.***paybutton***);  
 *// SqLite database handler* **db** = **new** SQLiteHandler(getApplicationContext());  
  
 *// session manager* **session** = **new** SessionManager(getApplicationContext());  
  
 **if** (!**session**.isLoggedIn()) {  
 logoutUser();  
 }  
  
 *// Fetching user details from SQLite* HashMap<String, String> user = **db**.getUserDetails();  
  
 String name = user.get(**"name"**);  
 String email = user.get(**"email"**);  
  
 **tv**.setText(Integer.*toString*(Integer.*parseInt*(Globals.*scredits*)));  
 **button**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **temp** = **editText**.getText().toString();  
 **int** a=Integer.*parseInt*(Globals.*scredits*);  
 **int** b=Integer.*parseInt*(**temp**);  
 **int** c=Integer.*parseInt*(Globals.*ccredits*);  
 **if**(b>a) {  
 Toast.*makeText*(Canteen.**this**, **"Not enough credits! Please Refill"**,  
 Toast.***LENGTH\_LONG***).show();  
 finish();  
 }  
 **else** {  
 a=a-b;  
 c=c+b;  
 Globals.*scredits*=(Integer.*toString*(a));  
 Globals.*ccredits*=(Integer.*toString*(c));  
 Toast.*makeText*(Canteen.**this**, **"Payment Successfull"**,  
 Toast.***LENGTH\_LONG***).show();  
 finish();  
 }  
 }  
 });  
  
  
 }  
 **private void** logoutUser() {  
 **session**.setLogin(**false**);  
  
 **db**.deleteUsers();  
  
 *// Launching the login activity* Intent intent = **new** Intent(Canteen.**this**, LoginActivity.**class**);  
 startActivity(intent);  
 *//finish();* }  
  
  
  
}

**CHAPTER 9: IMPLEMENTATION PLANNING**

**9.1 IMPLEMENTATION ENVIRONMENT (Single vs Multiuser, GUI vs Non GUI)**

The implementation environment is single user and has user friendly GUI in AD and AW.

**9.2 PROGRAM /MODULES SPECIFICATION**

In this project, we have used one database named android\_api having one table. The table is users and it contains name, email, uid, password, salt, created\_at, modified\_at, credits.

**9.3 CODING STANDARD**

* Class names have first letter in capital case.

Example: Login

* Variable names are in lower case.

Example: wpm

* Method names are in mixed case with first letter in lower case.

Example: void check();

* Name of packages are in lower case.

Example: package cashfree1test;

* Textarea and Textpane names are of the form name\_textarea.

Example: type\_textarea

* Buttons names are of the form name\_button. For example pause\_button.
* Buttons used to make the keyboard are named by letter name.

Example: a

**CHAPTER 10: TESTING**

**10.1 TESTING PLAN**

We have used Black box testing for the project.

1. **2 TEST SUITE DESIGN**
   * 1. **Test Cases**

**Register Function/ Module Test Suite and Test Case Design**

Test Suites No: 1

Test Suite Detail: Register Function with Correct Username and Password and Submit Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Register | Student ID: 14ce107  Email: abc@gmail.com  Password: Abc123 | This person is redirect to Successful toast  And back to homepage | This person is redirect to Successful toast and back to homepage | Pass |

Test Suites No: 2

Test Suite Detail: Register Function with Incorrect Username and Password and Submit Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Register | Student ID:  Email:  Password: | This person is redirect to Unsuccessful toast  And back to homepage | This person is redirect to Unsuccessful toast  And back to homepage | Pass |
| 2 | Register | Student ID: 14ce107  Email:  Password: Abc123 | This person is redirect to Unsuccessful toast  And back to homepage | This person is redirect to Unsuccessful toast  And back to homepage | Pass |
| 3 | Register | Student ID: varsh  Email: abc@gmail.com  Password: Varsh123 | This person is redirect to Unsuccessful toast  And back to homepage | This person is redirect to Unsuccessful toast  And back to homepage | Pass |

**Login Function/ Module Test Suite and Test Case Design**

Test Suites No: 1

Test Suite Detail: Login Function with Correct Username and Password and Submit Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Login | Student ID: 14ce107  Password: rani | This person is redirect to Use Credits page | This person is redirect to Use Credits page | Pass |

Test Suites No: 2

Test Suite Detail: Login Function with Incorrect Username and Password and Submit Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Login | Student ID:  Password: | This person is redirect to Unsuccessful toast | This person is redirect to Unsuccessful toast | Pass |
| 2 | Login | Student ID: 14ce107  Password: | This person is redirect to Unsuccessful toast | This person is redirect to Unsuccessful toast | Pass |
| 3 | Login | Student ID: abc@gmail.com  Password: Abc123 | This person is redirect to Unsuccessful toast | This person is redirect to Unsuccessful toast | Pass |

**Use Credits Module Test Suite and Test Case Design**

Test Suites No: 1

Test Suite Detail: Student Inquiry Details Function with Incorrect Information fill and Save Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Use\_Credits | Where:  Canteen OR Library OR Stationary  Amount: 100 | Successful transaction toast and database updated | Successful transaction toast and database updated | Pass |

**Use Credits Module Test Suite and Test Case Design**

Test Suites No: 2

Test Suite Detail: Student Inquiry Details Function with Correct Information fill and Save Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Use\_Credits | Where:  Canteen OR Library OR Stationary  Amount: | Unsuccessful toast : No amount entered | Unsuccessful toast : No amount entered | Pass |
| 2 | Use\_Credits | Where:  Canteen OR Library OR Stationary  Amount: 10000 | Unsuccessful toast : No credits available | Unsuccessful toast : No credits available | Pass |

**Show Credits Module Test Suite and Test Case Design**

Test Suites No: 1

Test Suite Detail: Student Inquiry Details Function with Incorrect Information fill and Save Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Show\_Credits | Show Click | This person is redirect to Show\_Credit page | This person is redirect to Show\_Credit page | Pass |

**Refill Credits Module Test Suite and Test Case Design**

Test Suites No: 1

Test Suite Detail: Faculty Details Function with Correct Information fill and Save Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Refill\_Credits | Admin ID: admin  Password: admin  Amount: 200 | Admin can update credit details in database | Admin can update credit details in database | Pass |

**Refill Credits Module Test Suite and Test Case Design**

Test Suites No: 2

Test Suite Detail: Faculty Details Function with Incorrect Information fill and Save Button Click.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Function Name | Test Case (condition) | Expected Results | Actual Result | Pass/Fail |
| 1 | Refill\_Credits | Admin ID: admin  Password: admin  Amount: | Unsuccesfull toast: Enter amount please. | Unsuccesfull toast: Enter amount please. | Pass |

**CHAPTER 11: LIMITATION AND FUTURE ENHANCEMENT**

Limitation:

* User cannot proceed transaction without login.
* Limited to Android Device only.

Future Enhancement:

* This app can be provided to Apple, Blackberry and other devices.
* More number of universities can be added up for use.

**CHAPTER 12: CONCLUSION AND DISCUSSION**

* 1. **SELF ANALYSIS OF PROJECT VIABILITIES**
* We thought that a student need not need to carry wallet and can still can get anything on campus through credits system.
* We also thought of android wear extension though the AD.
  1. **PROBLEM ENCOUNTERED AND POSSIBLE SOLUTIONS**
* There were so many problems related to database access via php in wamp server.
* Connecting wear to the AD was a major task too.
  1. **SUMMARY OF PROJECT WORK**
* We have developed a Student utility application for AD that can do all the transactions at CSA on campus.
* In Cash Free, the AD remaining transaction is being showed up in AW connected.