CHAPTER 1
INTRODUCTION

Time has always been a vital component in life of every person in this World and managing it is the most important task in current generation, where you have to be multi-tasking to survive and overcome the hurdles in life. Traditionally, this was done using binder/book but with time, it also needs to be updated.

Android Personal Scheduler is a perfect substitute for traditional binder/book Schedulers. You can create to-do lists, meetings, payments, store key information and passwords and mark important events using their integrated calendars.

From plain paper-and-binder Schedulers to personal information manager software for computers and smartphones, personal scheduler have come a long way. Designed to be portable, personal Schedulers usually include an address book, a diary to keep notes and a calendar to mark important events. Used for general and business purposes, a personal Scheduler helps users to set appointment dates, keep note of important events, create a to-do list and mark important dates to manage their day-to-day activities. Ever since personal Schedulers have gone digital, handling all kinds of personal tasks has become simpler and more streamlined. With the arrival of Android smartphones, people have chosen Android personal Scheduler applications over the PC versions as they are mobile and offer multi-platform support.

This Android Personal Scheduler application is the best way to manage your key business and personal information. You can simply create lists of tasks. You can store all key banks, credit card and contact information in separate categories designed for easy access in time of need.

Following tasks are performed by Android Personal Scheduler:-

1. To-Do: by using this option we can crate to do list. This application will alert the user at respective time.
2. Meeting: by using this option we can add meetings information and timings. This application will alert the user.
3. Payment: by using this option we can add payment information, the application will alert user at correct timing.

As useful as it is, the Android Personal Scheduler application is far from perfect at the moment but there’s a lot of scope in upgrading it. It doesn't save data locally, only to SD

card, you always need to be connected and signed in to use it and sometimes Android Personal Scheduler is very slow, taking ages to load notes. Hopefully these features will be addressed in future releases

Everything about Android Personal Scheduler is based around simplicity. The user interface of the application is clean and functional, making it easy for you to add and retrieve alerts. Buttons on the home screen allow you to add alerts.

The current Personal Schedulers available for android are completely stored on phones, which makes them unreliable in case of loss of phone, software reinstallation on phone and many other conditions. They are not trustworthy since they are not able to provide adequate security and reliability. Moreover these schedulers have a complicated User Interface for performing simple task which can be done easily.

Android Personal Scheduler has a Web Interface which can be used to add schedules and alerts are generated on phone via internet web service to the phone. It is highly secured because it requires user registration prior to the use of application. Android Personal Scheduler is also light weight and easy to use. This application runs on the latest android operating system i.e. Ice Cream Sandwich.

CHAPTER 2
LITERATURE REVIEW

A literature review discusses published information in a particular subject area, and sometimes information in a particular subject area within a certain time period. Literature reviews provide you with a handy guide to a particular topic. If you have limited time to conduct research, literature reviews can give you an overview or act as a stepping stone, [1].

2.1 Main Task of the Application

Android Personal Scheduler is basically a scheduler which stores the To-Do, Payment or Meeting Schedules and gives a reminder at a predefined time, [2]. The alerts are generated on the Mobile Device running Android Operating System at the instance of specified time.

Android Personal Scheduler stores 3 types of schedules, which are, To-Do List, Payments and Meetings. Alerts are generated on each type of scheduled activity at the given specific instance of time of the schedule. Adding a Schedule can also be done from internet web site from which the alerts will be generated to the phone via the web service under a condition that the internet connection is working, [2].

This can be beneficial for people who are busy and need reminders for their schedules wherever they are and whatever they are doing. For the busiest people their secretary or assistant can add schedules from the web site

2.2 Client-Server System

A computing system that is composed of two logical parts: a server, which provides services, and a client, which requests them. The two parts can run on separate machines on a network, allowing users to access powerful server resources from their personal computers, [2].

The server portion almost always holds the data, and the client is nearly always responsible for the user interface. The application logic, which determines how the data should be acted on, can be distributed between the client and the server, [3].

The client/server model has become one of the central ideas of network computing. Most business applications being written today use the client/server model. So does the Internet's main program, TCP/IP. In marketing, the term has been used to distinguish distributed computing by smaller dispersed computers from the "monolithic" centralized computing of mainframe computers. But this distinction has largely disappeared as

mainframes and their applications have also turned to the client/server model and become part of network computing.

2.3 Existing System

Strategic planning fails is an assertion that strategic planning fails in the first place and that the causes might not be well understood. The reasons why strategic planning fails and the rate of failures should not be a surprise to anyone since it would perpetuate another fallacy of planning, [4]. Organizations devote so much time and energy to the annual ritual of developing or updating their strategic plans if they are doomed to failure from the start. Just think of all of the smart, well-educated, experienced managers and executives devoting so much of their organizations limited resources on an activity with such a poor track record of producing results.

The Existing Schedulers are capable of generating the alerts for the scheduled activities but they use the inbuilt storing facilities of the phone only to keep records of the schedules. Hence, in case of phone formatting or loss of phone there is no option for retrieving the schedules stores as they are not backed up anywhere. This makes the existing schedules unreliable, [5].

The schedules can be added or deleted from the phone only from its user interface. Hence, you assistant or secretary has to add all the schedules at once, [6]. Modifying or deleting the schedules is not possible once you are alone performing any task. Moreover the User Interfaces of the current systems are more complicated and that can also be simplified.

2.4 Proposed System

The proposed system contains a web site on server which will be used to add schedules to the database located on the server. These schedules will generate alert on the mobile device via web service which will be used as the medium of communication between the server and the phone. Hence, in case of phone formatting or loss of phone the schedules will still be retrievable and will still generated alerts on new devices once logged in from there, [7].

The user will also be able to add schedules to the android application from its interface. These will also generate alerts on the specified time. The alerts will be in the forms of alarm which will be invoked if a schedule is found, [8].

The User Interface of the proposed system will be easy, which will reduce the learning curve for the new user. The application will also be light weight yet powerful in performing its functions, [9].

The proposed System will be able to store following types of the schedules

1. To-Do: by using this option we can crate to do list. This application will alert the user at respective time.
2. Meeting: by using this option we can add meetings information and timings. This application will alert the user.
3. Payment: by using this option we can add payment information .the application will alert user at correct timing.

The architecture of the proposed system is as follows.

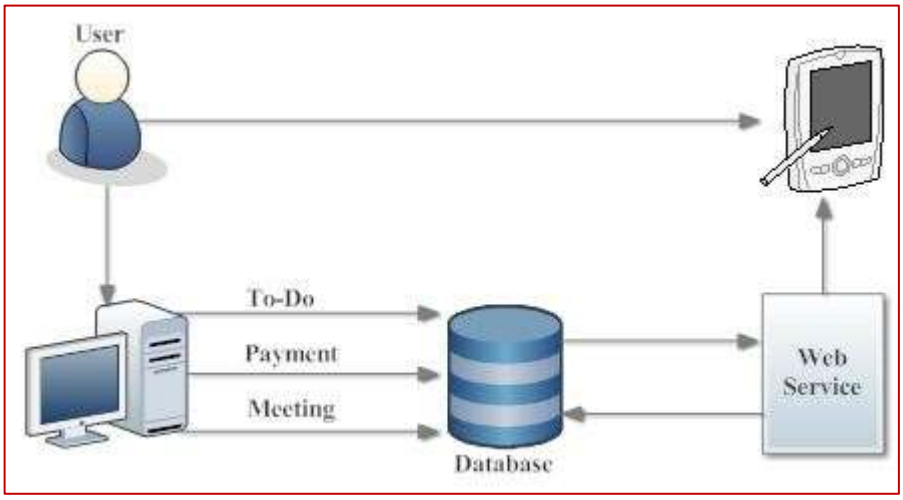


Figure 2.1: Architecture of Proposed System

2.4.1 Advantages:

- Manage all your reminders.
- Create, Modify and Delete your schedules.
- Generate and fulfill your ideas.
- Plan your time.
- Make your workplace more comfortable.

CHAPTER 3
SYSTEM ANALYSIS

The analysis of the requirements of a task and the expression of those requirements in a form that permits the assembly of computer hardware and software to perform the task is called as the System Analysis. It is done in the following 3 steps:-

- Technical Feasibility
- Economic Feasibility
- Operation Feasibility

3.1 Technical Feasibility

The proposed scheduler system uses all the currently used technical hardware and software which are being used by the current scheduler systems. Simple computers can be found anywhere and once signed up you don't compulsorily need a computer system to operate the application.

Android Phones running Ice-Cream Sandwich are being used by zillions right now and the number is still increasing. Hence, it is completely feasible for this project to be implemented.

3.2 Operation Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. Android Personal Scheduling is proving successful in all the fields, and will overcome the recognized problems at current stage. Hence, it is feasible in this category too.

3.3 Economic Feasibility

Economic feasibility of this project is also possible as most of the systems being used for this project are already being used by the existing systems. The only price increase in this project is the price required for the servers space used for storing the schedules from thee internet. All other requirements are similar to the requirements of the existing systems.

3.4 Requirements

3.4.1 Software Requirement:

Operating system : - Microsoft XP/ W7/ 2008 server/ W8 or Above

Front End (Platform)	: - Visual Studio 2010 or Above, Eclipse Indigo
Framework	: - .NET3.5, .NET4.0
Back end (Database)	: - MS SQL Server 2012, MY SQLITE
Android Mobile OS	: - Ice-cream Sandwich or Above

3.4.2 Minimum Hardware requirements:

Processor	: - Pentium IV or Above
RAM	: - 1 GB RAM or Above
Mobile Device	: - Any Mobile Device Running Android 4.0 or Above

3.5 What Is Microsoft .NET?

This is how Microsoft describes it: ".NET is the Microsoft Web services strategy to connect information, people, systems, and devices through software. Integrated across the Microsoft platform, .NET technology provides the ability to quickly build, deploy, manage, and use connected, security-enhanced solutions with Web services. .NET connected solutions enable businesses to integrate their systems more rapidly and in a more agile manner and help they realize the promise of information anytime, anywhere, on any device."

Microsoft .NET is Microsoft's new Internet strategy:-

- .NET is a server centric computing model.
- .NET is a new Internet and Web based infrastructure.
- .NET delivers software as Web Services.
- .NET is a framework for universal services.
- .NET will run in any browser on any platform.
- .NET is based on the newest Web standards.
- .NET is NOT a new OS, DB or PL

3.6 .NET Framework Overview

The .NET Framework is Microsoft's platform for building applications that have visually stunning user experiences, seamless and secure communication, and the ability to model a range of business processes. The .Net Framework consists of:-

- Common Language Runtime – provides an abstraction layer over the operating system
- Base Class Libraries – pre-built code for common low-level programming tasks
- Development frameworks and technologies – reusable, customizable solutions for larger programming tasks

By providing you with a comprehensive and consistent programming model and a common set of APIs, the .NET Framework helps you to build applications that work the way you want, in the programming language you prefer, across software, services, and devices.

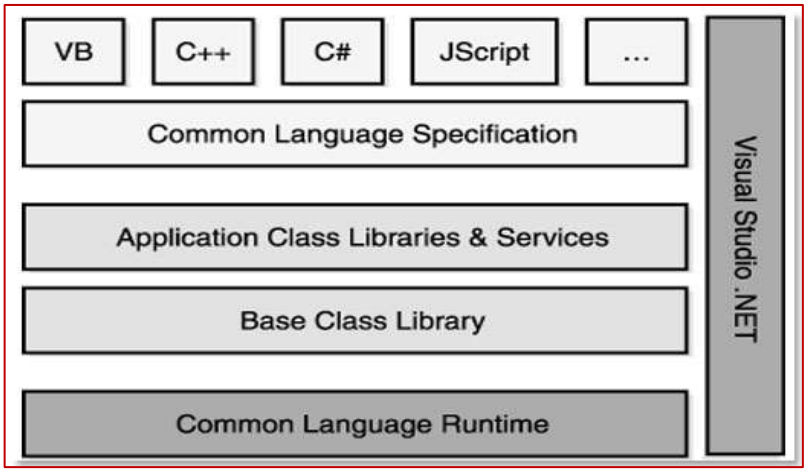


Figure 3.1: .NET Framework

3.6.1 Secure, Multi-Language Development Platform.

Developers and IT professionals can count on .NET as a powerful and robust software development technology that provides the security advancements, management tools, and updates you need to build, test, and deploy highly reliable and secure software. .NET provides a multi-language development platform, so you can work in the programming language you prefer. The Common Language Runtime (CLR) provides support for powerful, static languages like Visual Basic© and Visual C#, and the advent of the Dynamic Language Runtime (DLR) means that dynamic languages, such as Managed Jscript, Iron Ruby and Iron Python, are also supported.

3.6.2 Cutting-Edge Web Application Development.

ASP.NET is a free technology that enables Web developers to create anything from small, personal Web sites through to large, enterprise-class dynamic Web applications. Microsoft's free AJAX (Asynchronous JavaScript and XML) framework – ASP.NET AJAX – enables developers to quickly create more efficient, more interactive, and highly personalized Web experiences that work across all of the most popular browsers.

3.6.3 Secure, Reliable Web Services.

For service-oriented programming Windows Communication Foundation (WCF) unifies a broad array of distributed systems capabilities in a compassable and extensible architecture,

spanning transports, security systems, messaging patterns, encodings, network topologies, and hosting models.

3.6.4 Flexible Data Access Options.

ADO.NET is a set of classes that expose data access services to the .NET programmer. ADO.NET provides a rich set of components for creating distributed, data-sharing applications. It is an integral part of the .NET Framework, providing access to relational, XML, and application data. ADO.NET supports a variety of development needs, including the creation of front-end database clients and middle-tier business objects used by applications, tools, languages, or Internet browsers.

ADO.NET Entity Framework simplifies application data access by providing an extensible, conceptual model for data from any database and enables this model to closely reflect business requirements.

3.7 Android (Operating System)

The Android platform is packing some serious heat these days in the mobile marketplace and gaining traction worldwide. The platform has seen numerous advancements in terms of SDK functionality, handset availability, and feature set, [10]. A wide diversity of Android handsets and devices are now shipping and in consumer’s hands and we are not just talking about phones, Android has begun to ship on netbooks, Internet tablets (such as the Nexus 7), eBooks readers (like the Barnes & Noble nook), digital photo frames, and a variety of other consumer electronics. There are even proof-of-concept appliances such as an Android microwave and washer/dryer combo.

Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance, a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. The first Android-powered phone was sold in October 2008.

3.7.1 Android 4.0–4.0.2 Ice Cream Sandwich (API level 14)

The SDK for Android 4.0.1 (Ice Cream Sandwich), based on Linux kernel 3.0.1, was publicly released on 19 October 2011. Google's Gabe Cohen stated that Android 4.0 was "theoretically compatible" with any Android 2.3.x device in production at that time. The source code for Android 4.0 became available on 14 November 2011.

3.7.2 Android Software Development

3.7.2.1 Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later. The officially supported integrated (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and Net Beans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executable), resource files, etc.

3.7.2.2 Eclipse Indigo

In computer programming, Eclipse is a multi-language software development environment comprising a base workspace and an extensible in system for customizing the environment. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plugins, other programming languages including Ada.

3.8 SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time). Its primary query languages are T-SQL and ANSI SQL.

3.8.1 SQL Server 2012

At the 2011 Professional Association for SQL Server (PASS) summit on October 11, Microsoft announced that the next major version of SQL Server (codenamed "Denali"), would be SQL Server 2012. It was released to manufacturing on March 6, 2012.

It was announced to be the last version to natively support OLE DB and instead to prefer ODBC for native connectivity.

SQL Server 2012's new features and enhancements include Always On SQL Server Failover Cluster Instances and Availability Groups which provides a set of options to improve database availability, Contained Databases which simplify the moving of databases between instances, new and modified Dynamic Management Views and Functions, programmability enhancements including new spatial features, metadata discovery, sequence objects and the THROW statement, performance enhancements such as Column Store Indexes as well as improvements to Online and partition level operations and security enhancements including provisioning during setup, new permissions, improved role management, and default schema assignment for groups.

3.8.2 SQLite

SQLite is a relational database management system contained in a small (~350 KB) C programming library. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it.

SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain.

SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. SQLite has many bindings to programming languages.

The source code for SQLite is in the public domain. It is a native storage system in Android.

CHAPTER 4
SYSTEM DESIGN

This chapter explains graphically how the proposed project is going to work with the help of graphical diagrams. For better understanding purpose it has been divided into 3 different parts, which are as follows:-

- Data Flow Diagram
- Flow Charts
- Database Schema

4.1 Data Flow Diagram

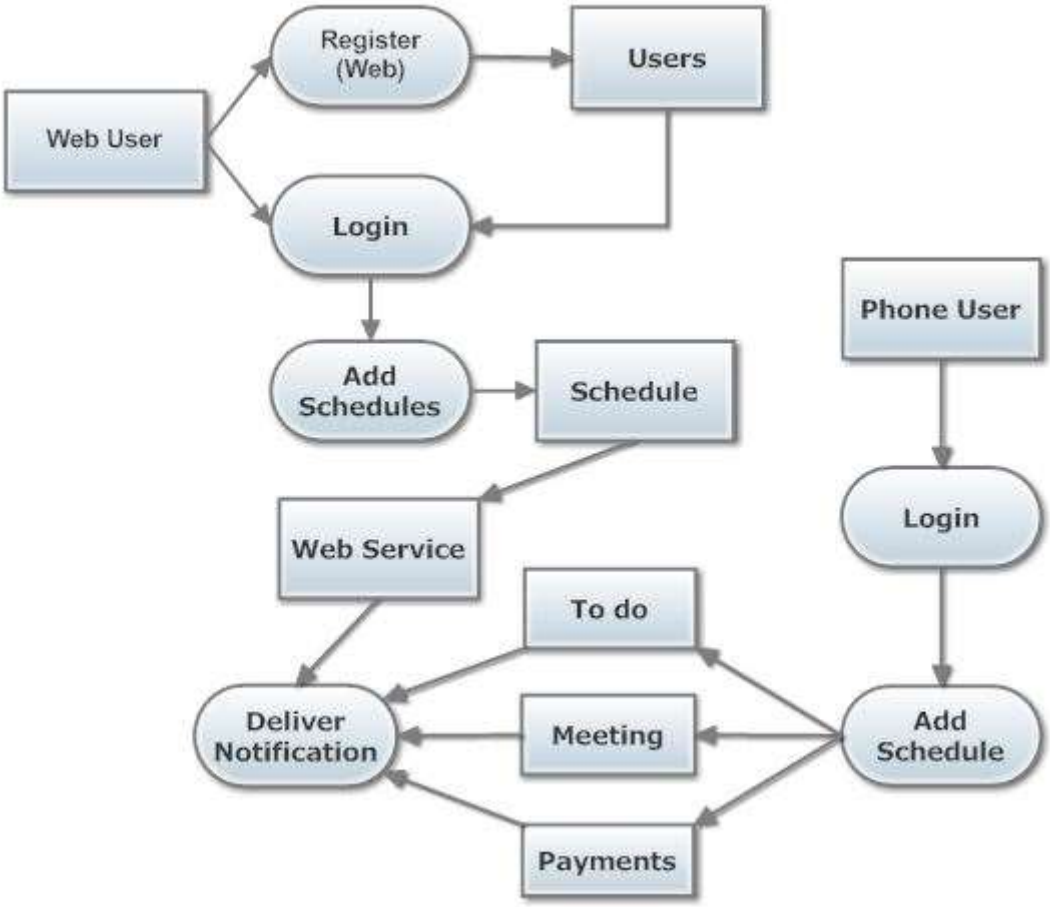


Figure 4.1: Data Flow Diagram

This is the complete data flow diagram of the project which explains how the complete project is going to work from all points of view, i.e. from the view of web user as well as the view of the phone user.

4.2 Flow Charts

4.2.1 Login Page

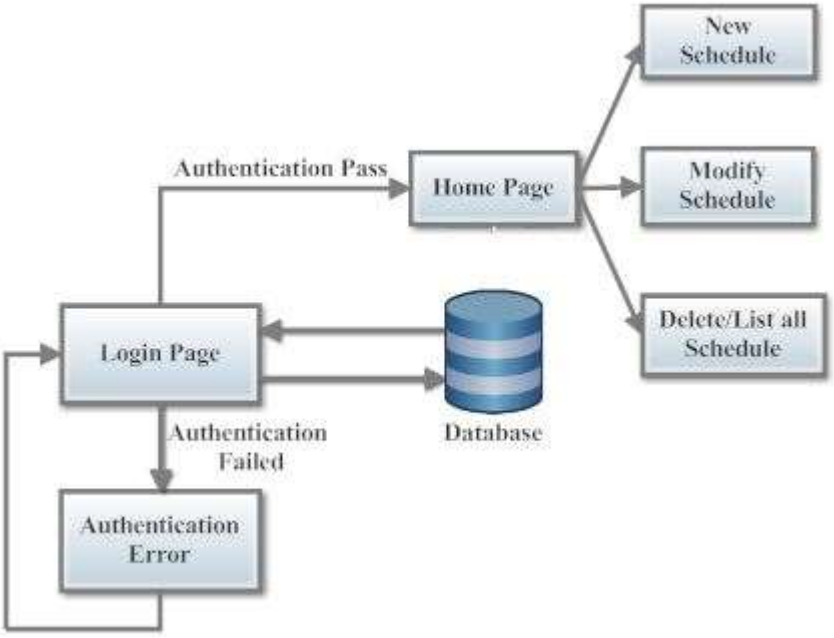


Figure 4.2: Flow Chart for Login Page

This figure shows the flow chart for the login page of the project. It explains the details for both the condition, which are, authentication pass and authentication pass.

4.2.2 New Schedules

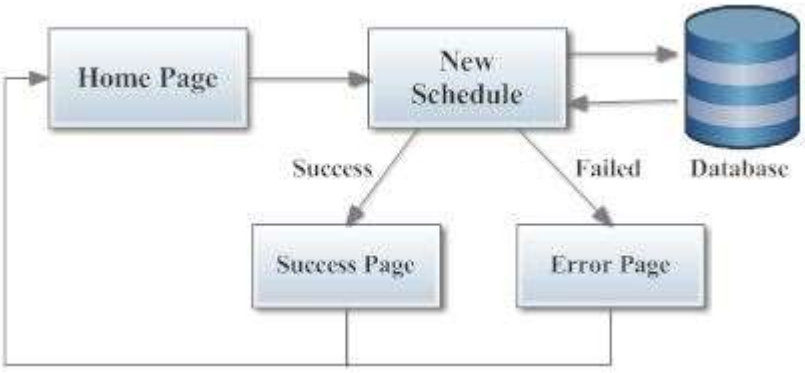


Figure 4.3: Flow Chart for New Schedule

This figure shows the flow chart for the Adding Schedule. It explain how a new schedule is added to the database.

4.2.3 Modify Schedule

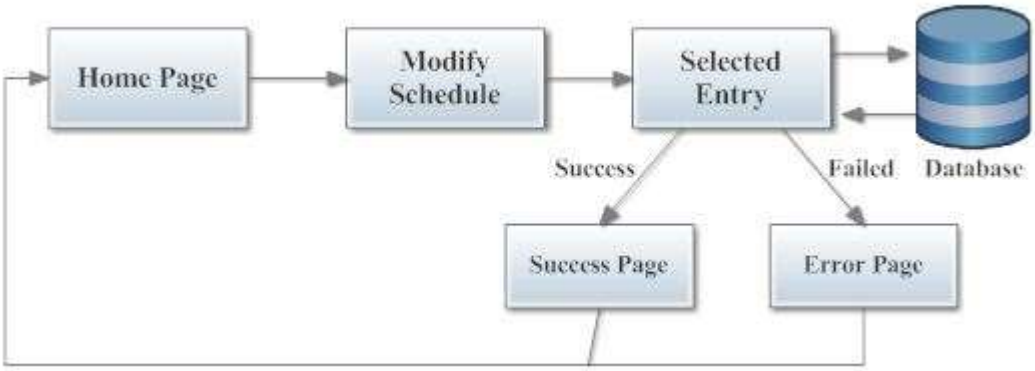


Figure 4.4: Flow Chart for Modify Schedule

This figure shows the flow chart for modifying a schedule. It explains how a schedule is modified and stored again in the database.

4.3 Database Schema

4.3.1 Database Schema for User Login Details

Table 4.1: Master Login

Sr No	Field	Type	Null	Primary Key
1	Userid	Varchar(100)	No	Yes
2	Usernm	Varchar(100)	No	-
3	Pass	Varchar(100)	No	-
4	Sques	Varchar(100)	No	-
5	Sans	Varchar(100)	No	-

This table shows the database schema for the table used to store the Login details of the users. This table also stores the information necessary for the user identification in case of password retrieval when the user has forgotten or lost the password.

This table contains five columns all of varchar type with a character limit of 100. None of the column accepts null value.

4.3.2 Database Schema for Schedules

Table 4.2: Schedules

Sr No.	Field	Type	Null	Primary Key
1	sid	Int	No	Yes
2	schtype	Varchar(80)	No	-
3	date	Int	No	-
4	month	Int	No	-
5	year	Int	No	-
6	activity	Varchar(300)	No	-
7	remember	Varchar(200)	No	-
8	status	Varchar(80)	No	-
9	time	Varchar(100)	No	-
10	userid	Varchar(100)	No	-

This table is used to store all the schedule details. It contains ten columns, out of which, four are of integer type and remaining are of varchar type. None of these accept null value as input.

CHAPTER 5
IMPLEMENTATAION

This chapter shows the practical implementation of the Android Personal Scheduling. For understanding Purpose it has been divided in 3 main sections

- Web Module
- Web Service
- Android Application

The Web Module contains the screenshots from the web site, The Web Service contains the screenshots from the web service used to connect the phone and the web database and the Android Application includes the screenshots of the android application running on a phone.

5.1 Web Module

5.1.1: Login Page

Figure 5.1: Login Page

This is a Login page which will take us to User Home Page, provided that User-ID and password is correct, otherwise, it will send us to page displaying Error Message and we can also register if you are new user.

5.1.2 Password Recovery

Figure 5.2: Password Recovery

This page is used to recover a user password in case of the password lost or forgotten. It requests the user for 3 things User ID, Security Question and Security Answer. If everything provided is correct then a random password is generated for temporary purpose.

5.1.3 New User Registration

Figure 5.3: Registration Page

This page is used to create a new account of user. It contains various fields asking the Details for creating a new account. It contains user name, user Id, Password, Retype Password, Security Question and Answer.

5.1.4 Registered successfully

Figure 5.4: Registration Successful

This page is displayed when new user has successfully completed registration process. It indicates that your user is successfully registered and we can go to home.

5.1.5 User Home Page

Figure 5.5: User Home Page

This is the User home Page. It shows all the options available for the users, which are, new schedule, modify my schedules and my schedules. In new schedule user can add new schedules. Modify Schedule allows user to modify existing schedules. In my schedules we can check detail of schedule which have been inserted before or delete them.

5.1.6 New Schedule Entry

Figure 5.6: New Schedule Entry

This page is used to insert new schedule entry. Here we have to select schedule type, date, time, activity and time to reminder before. All these things are compulsory and new schedule is not created if any of these are not given.

5.1.7 Information Stored Successfully

Figure 5.7: Information Stored Successfully

This page shows that you have successfully added a new schedule to the database and it includes a home link to go back to the user home page.

5.1.8 Modify Schedules

Figure 5.8: Modify Schedules

This page shows complete list of schedules added to the database. Clicking on the "Modify" link in the first column of the table allows user to modify that respective entry.

5.1.9 Modify Schedule Entry

Figure 5.9: Modify Schedules Entry

This page is used to modify schedule entry which is selected and we can modify schedule type, date, time, activity and remember before time entered at the time of inserting the schedule.

5.1.10 Delete schedules

Figure 5.10: Delete Schedules

This page shows all entry inserted by user and we can delete one or multiple schedules from the list by selecting the checkbox in the first column of the table and pressing delete. To delete multiple entries, select multiple checkboxes.

5.1.11 Change Password Page

Figure 5.11: Change Password Page

This page is used to change login user password. To change login password we need to insert current password, new password and retype new password for security purpose. The password is only changed if and only if the string entered in the New Password and Retype Password are matching.

5.2 Web Service

5.2.1 MyScheduler

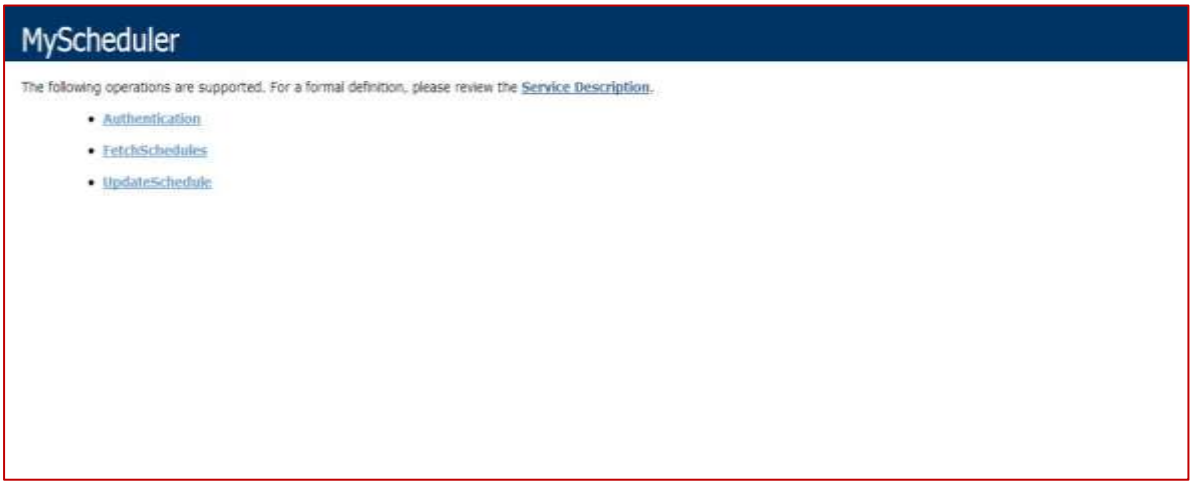


Figure 5.12: MyScheduler

This is MyScheduler page. It shows all the methods of the web service, which are, Authentication, Fetch Schedules and update schedule.

5.2.2 Authentication Page

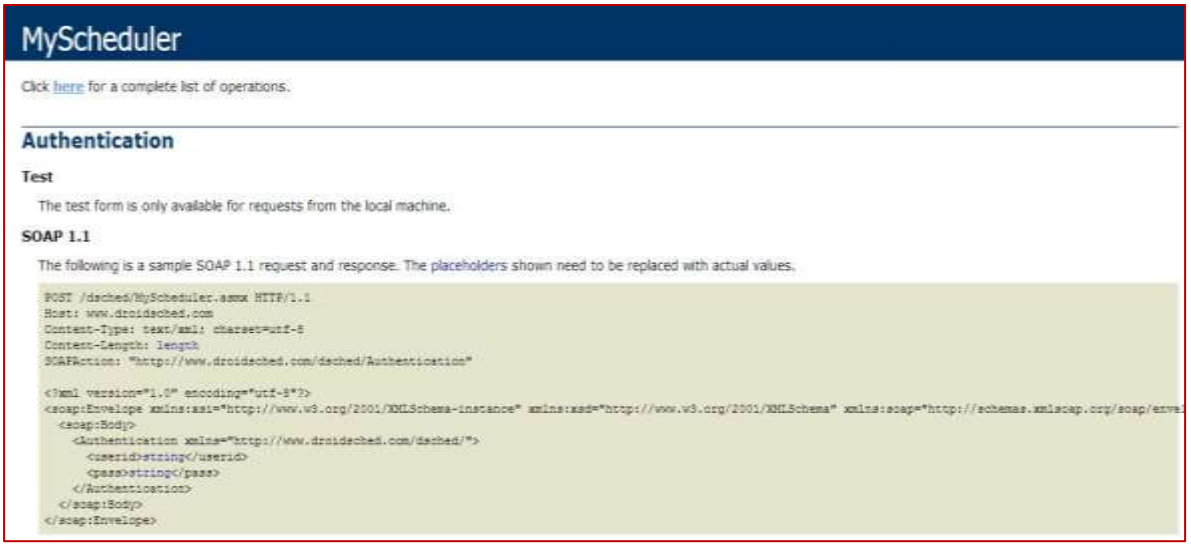


Figure 5.13: Authentication

This is Authentication method of web service. This method of the web service takes two comments, user id and password and returns true if they are correct or false if they don't match. This is used to login from the phone.

5.2.3 Fetch Schedules Page

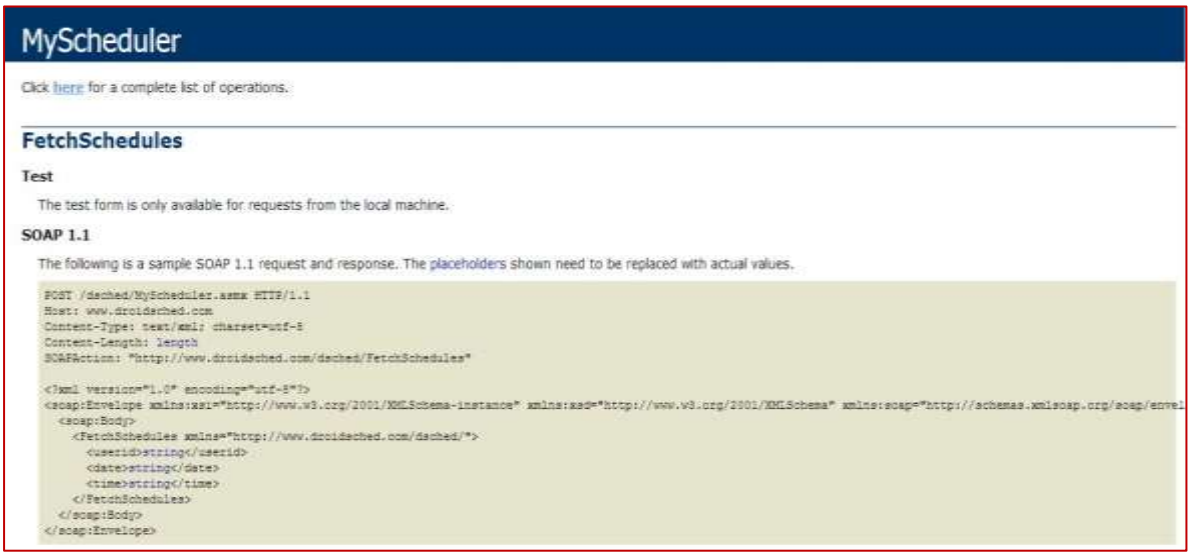


Figure 5.14: Fetch Schedules Page

This is Fetch Schedules method. This method of web service takes three comments, user ID, Date and Time at the instance. If a schedule is found then it returns the id and the name of schedule or it returns NA, by default, if no schedule is found.

5.2.4 Update Schedules Page

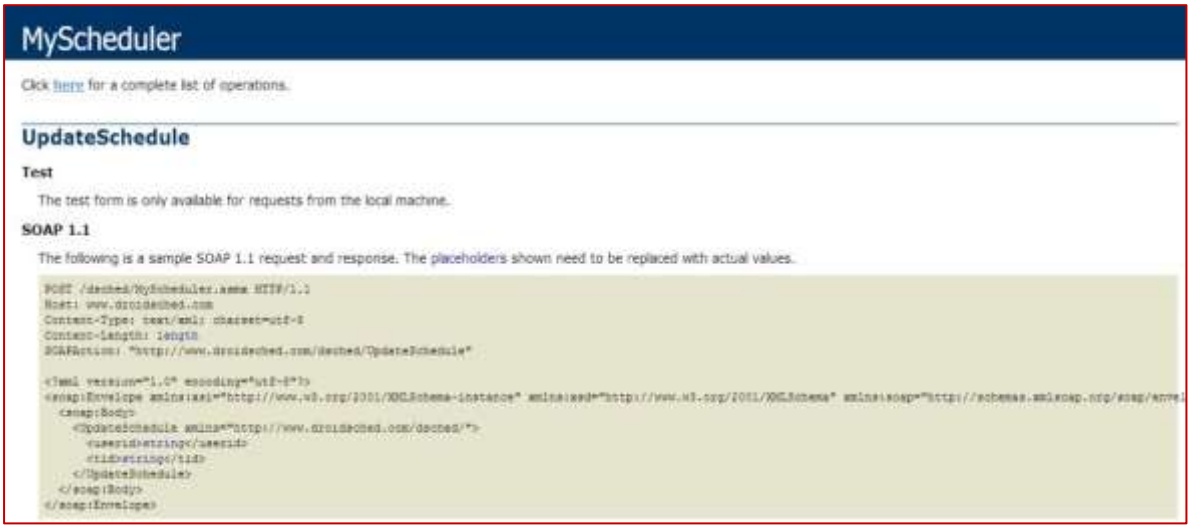
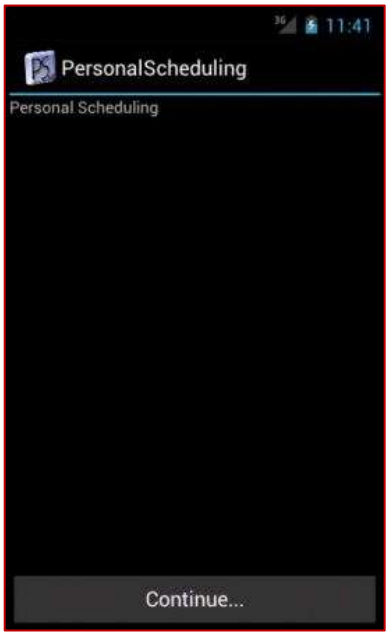


Figure 5.15: Update Schedules Page

This is Update Schedules method. This method of web service takes user ID and the Alarm ID so that the alarm status is updated in the database located on the server for the schedules entered via the Internet.

5.3 Android Application

5.3.1 Splash Screen



5.16: Splash Screen

This is splash screen of the android application which is used to go to the login page.

5.3.2 Login Page

5.17: Login Page

This is a Login page which will take us to either Home page or depending on User-Id and password is correct otherwise it will send us to page displaying Authentication failed. And we can also register if you are new user.

5.3.3 Home Page

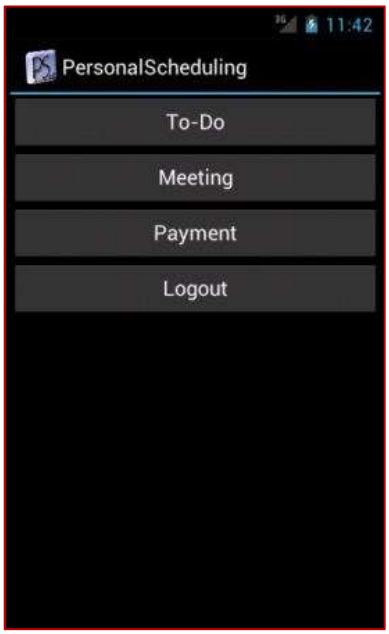


Figure 5.18: Home Page

This page shows To-Do, Meeting, Payment and Logout. In To-Do, Meeting and Payment we can check the detail of schedule which has insert before and can insert new detail.

5.3.4 Manage To-Do

Figure 5.19: Manage To-Do

This page is used for insert add new schedule, modify old schedule, list of schedule, go back to main menu and Exit.

5.3.5 To-Do Entry

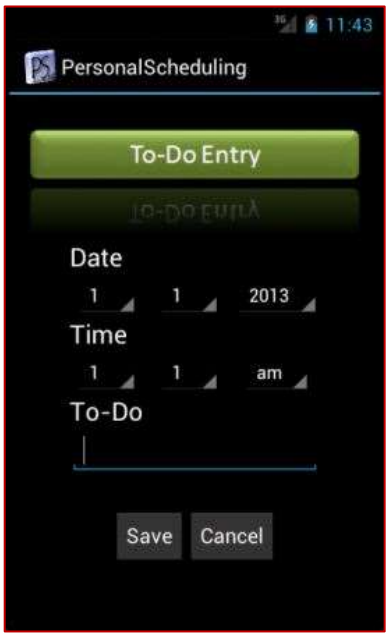


Figure 5.20: To-Do Entry

This page is used to insert new To-Do schedule entry. Here we can select Date, Time and Activity.

5.3.6 To-Do List

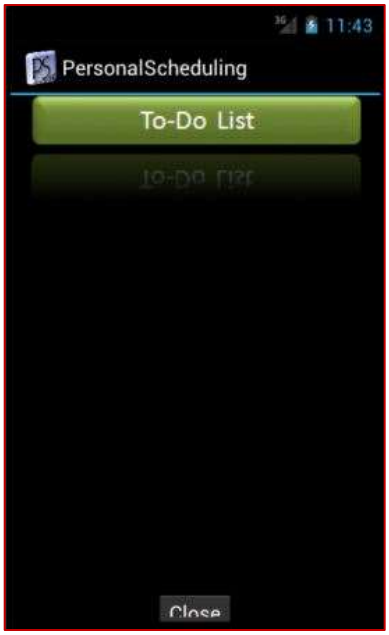


Figure 5.21: To-Do List

This page is used for view list of To-Do schedule entry.

5.3.7 Manage Meeting

Figure 5.22: Manage Meeting

This page is used to manage payment for insert add new schedule, modify old schedule, list of schedule, go back to main menu and Exit.

5.3.8 Manage Payment

Figure 5.23: Manage Payment

This page is used to manage payment for insert add new schedule, modify old schedule, list of schedule, Exit and go back to main menu.

CHAPTER 6
SOFTWARE TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs, [11].

Software Testing can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development works as expected and can be implemented with the same characteristics.

6.1 The Box Approach

Software testing methods are traditionally divided into Black box Testing and White Box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

6.1.1 Black Box Testing

Black box Testing treats the software as a "black box"-without any knowledge of internal implementation. It includes specification-based testing.

6.1.1.1 Specification-based testing

Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks.
The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs.

Using the principle, "Ask and you shall receive", black box testers find bugs where programmers do not. But, on the other hand, black box testing has been said to be "like a

walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when

- A tester writes many test cases to check something that could have been tested by only one test case.
- Some parts of the back-end are not tested at all. Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

6.1.2 White Box Testing

White Box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

The following types of white box testing exist:-

- Fault injection methods - improving the coverage of a test by introducing faults to test code paths.
- Test coverage.

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:-

- Function coverage, which reports on functions executed.
- Statement coverage, which reports on the number of lines executed to complete the test.

They both return code coverage metric, measured as a percentage.

6.2 Testing Levels

Tests are frequently grouped by where they are added in the software development process, or by the level of specificity of the test.

6.2.1 Unit Testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might

have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other. Unit testing is also called Component Testing.

6.2.2 Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed and works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

6.2.3 System Testing

System testing tests a completely integrated system to verify that it meets its requirements. System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

6.2.4 System Integration Testing

System integration testing verifies that a system is integrated to any external or third party systems defined in the system requirements. System Integration testing (SIT) is a testing process that exercises a software system's coexistence with others. With multiple integrated systems, assuming that each have already passed system testing, SIT proceeds to test their required interactions.

6.2.5 Regression Testing

Regression testing focuses on finding defects after a major code change has occurred. Specifically, it seeks to uncover software regressions or old bugs that have come back. Such regressions occur whenever software functionality that was previously working correctly stops working as intended. Typically, regressions occur as an unintended consequence of program changes, when the newly developed part of the software collides with the previously existing code. Common methods of regression testing include re-running previously run tests and checking whether previously fixed faults have re-emerged.

CHAPTER 7
ADVANTAGES

The advantages of Android Personal Scheduler are as follows:-

- Keep your tasks, projects and contacts always at hand.
- Get prompt reminders for all schedules.
- Schedules added from internet are stored on server for retrieving again.
- Simple User Interface
- Light Weight yet powerful.

CHAPTER 8
FUTURE SCOPE

There are numerous possibilities for future development in this project which can be explored but the most prominent ones are listed below:-

- **Get SMS Alerts From Gate-Way Services.**
  This will make sure that the user gets alerts for all the schedules even if they are not connected to the internet for the whole time. The alerts will be sent to the registered number of the user.
- **Add Personal Notes Or Text.**
  Users will be able to add notes and other text details to each of their schedules for more details about the event. This will help the users to be more organized and well informed about the events and schedules.
- **Send Scheduled SMS Or Mail.**
  This feature will allow the user to schedule an SMS or Mail for the future which will be automatically sent to the required receiver.
- **Add Personal Files (Document, Audio, Image, Video Etc.).**
  This will allow users to store personal documents in various formats for easy and reliable access.

CHAPTER 9
CONCLUSION

Android Personal Scheduling application can prove to be beneficial for all the people to manage their daily life and it also has all the necessary features and back-up plan for most of the extreme condition that one might face.

Android Personal Scheduling can also prove to be beneficial for people whose schedules are managed by their secretary or personal assistant. These people can manage their schedules using the online service and those will be updated on the go.

Thus, Android Personal Scheduling can be used by anyone to reduce their stress over time management and to avoid situation when you have to be at two places at the same time or who are addicted to forget important events.