**ABSTRACT**

Any mobile user using this project is aimed at developing an application that incorporates both location and data call technology, steering you towards the very best activities that your city has to offer. For the selected category, the application retrieves appropriate list of options for any location review. The application provides the full information (address, contact number and other information) for any selected option. User also has an option to call the selected option or SMS the info from the application.

The **City Search** is an online guide that provides information about businesses in the categories of dining, entertainment, retail, travel, and professional services in cities. Here the end user would be a Mobile user as it’s an Android technology; so that the user can finally implement all the needed operations in their own mobile device. The Mobile User should be able to search appropriate location. The Mobile User should be able to retrieve the very best activities for the selected city. The user can also be able to retrieve full information (address, contact number and other information) for the any selected option. The Mobile User should also have an option to call the selected option or SMS the information from the application.

1

**Contents**

Abstract

List of figures

List of tables

**Chapter 1**

**Introduction**

1.0 Introduction to this chapter

1.1 Objective

1.2 Scope

1.3 Technology

1.4 Overview

**Chapter 2**

**Analysis**

2.0 Introduction

2.1 Usecase

2.2 Test cases

2.3 SRS

2

Chapter 3

**Design**

3.0 Introduction

3.1 Usecase diagram

3.2 Sequence diagram

3.3 Activity diagram

Chapter 4

**Implementation Technology**

4.0 Introduction

4.1 Modules and description

Chapter 5

**Testing and Screens**

5.0 Introduction

5.1 Testing

5.2 Test Results and Screens

5.3 Conclusion

3

**Screens**

1.Screen1……………………………………………………Home Page

1. Screen2……………………………………………………City
2. Screen3……………………………………………………Cities list
3. Screen4……………………………………………………Category
4. Screen5……………………………………………………Categories list
5. Screen6…………………………………………………….Locations
6. Screen7……………………………………………………Available Locations
7. Screen8……………………………………………………Location details
8. Screen9……………………………………………………Select to call
9. Screen10…………………………………………………Call processing
10. Screen11…………………………………………………Dialing number
11. Screen12…………………………………………………Email details
12. Screen13…………………………………………………Mailing
13. Screen14…………………………………………………Message saved
14. Screen15…………………………………………………Message discarded

4

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Figure No** | **Figure Name** | **Page No** |
| 1 | 3.1.0 | Use Case Diagram | 30 |
| 2 | 3.2.0 | Sequence Diagrams | 33 |
| 3 | 3.3.0 | Activity Diagram | 36 |

5

CHAPTER-1

**INTRODUTION**

**1.0 INTRODUCTION**

**1.1 OBJECTIVE**

**The** City Search is an online guide **that** provides information about businesses in thecategories of dining, entertainment, retail, travel, and professional services in cities.

**1.2 SCOPE**

The Mobile User should be able to search appropriate location. The Mobile User should be able to retrieve the very best activities for the selected city. The user can also be able to retrieve full information (address, contact number and other information) for the any selected option. The Mobile User should also have an option to call the selected option or SMS the information from the application.

**1.3 TECHNOLOGY**

**SYSTEM REQUIREMENTS**

Front end: j2me

Back end: java DB

Tools : Net beans, Glassfish IDE with android plug-in, android SDK1.5

6

**HARDWARE REQUIREMENTS**

**1.** **Development environment**

a. Hard disk: 160 GB

b. RAM : 256 MB

1. **Application environment**

a. Mobile with android O.S or java supported

**1.4 OVERVIEW**

City search is an online guide that provides information about businesses in the categories like

Dining, Entertainment, Travel, Professional services, etc

**JAVA**

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web application.

7

The original and reference implementation Java compilers, virtual machines and class libraries were developed by Sun from 1995. Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and super computers on the high end. Java is used in mobile phones, Web servers and enterprise applications, and while less common on desktop computers; Java applets are often used to provide improved and secure functionalities while browsing the World Wide Web.

Writing in the Java programming language is the primary way to produce code that will be deployed as Java byte code, though there are byte code compilers available for other languages such as JavaScript, Python, and Ruby. Several new languages have been designed to run natively on the Java Virtual Machine (JVM) Java eliminates certain low-level constructs such as pointers and has a very simple memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.

**JAVA DERBY**

Java DB is Sun's supported distribution of the open source Apache Derby 100% Java technology database. It is fully transactional, secure, easy-to-use, and standards-based.

The features are

Full-featured, easy-to-use, SQL database.



Protection against data corruption and system crashes.



Free under the Apache license.



Runs everywhere from PDAs to laptops to mainframes.



8

Embeddable in applications.



Portable and 100% Java, CDC, SQL, and JDBC compliant.



Easy, one-click installation.



Small footprint of 2.5 MB.



**NET BEANS IDE**

**Net Beans** (IDE) refers to both a platform framework for Java desktop applications, andan integrated development environment. The Net Beans IDE is written in Java and runs everywhere where a JVM is installed, including Windows, Mac OS, Linux, and Solaris. A JDK is required for Java development functionality, but is not required for development in other programming languages.

The Net Beans Platform allows applications to be developed from a set of modular software components called *modules*. Applications based on the Net Beans platform (including the Net Beans IDE) can be extended by third party developers. Net Beans began in 1996. Sun open-sourced the Net Beans IDE in June 1999. Net Beans IDE 6.5, released in November 2008, extended the existing Java EE feathers (including Java Persistence support, EJB 3 and JAX-WS). Additionally, the NET Beans Enterprise Pack supports development of Java EE 5 enterprise applications, including SOA visual design tools, XML schema tools, web services orchestration (for BPEL), and UML modeling.

**NETBEANS PLATFORM**

The **Net Beans Platform** is a reusable framework for simplifying the development of Java Swing desktop applications. The Net Beans IDE bundle for Java SE contains what is needed to start developing Net Beans plugins and Net Beans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally-signed upgrades

9

and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

User interface management (e.g. menus and toolbars)



User settings management



Storage management (saving and loading any kind of data)



Window management



Wizard framework (supports step-by-step dialogs)



Net Beans Visual Library



The **Net Beans IDE** is an open-source integrated development environment. Net Beans IDE supports development of all Java application types (Java SE including JavaFX, (Java ME, web, EJB and mobile applications) out of the box. **Modularity**: All the functions of the IDE are provided by modules. Each module provides a well defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the Net Beans IDE.

10

CHAPTER-2

**ANALYSIS**

**2.0 Introduction**

The major objectives of systems analysis are to find answers for each business process:

What is being done, How is it being done, who is doing it, When is he doing it, Why is it being done and How can it be improved? It is more of a thinking process and involves the Creative skills of the System Analyst.

**2.1 USECASE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Source** |  |  |  |  |
| **S. No** | **Requirement ID** | **Requiremen** | **(user/admi** | **Stable** | **Priority** | **Need** | **Verifiable** |
|  |  | **t name** | **nistrator)** | **(y/n)** |  |  | **(y/n)** |
|  |  |  |  |  |  |  |  |
| 1 | CS\_UC\_01 | City search | User | Y | High | Essential | y |
|  |  |  |  |  |  |  |  |
| 2 | CS\_UC\_02 | View city | User | Y | High | Essential | y |
|  |  |  |  |  |  |  |  |
| 3 | CS\_UC\_03 | View | User | Y | High | Essential | y |
|  |  | categories |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 4 | CS\_UC\_04 | View | User | Y | High | Essential | y |
|  |  | information |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Requirements look up table**

11

**2.2 Test cases**

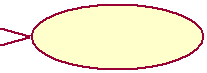
A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle... Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites.

12

**The Test Cases**

**Test case1**

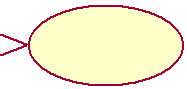
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID | CS\_UC\_00 | | | | |  |  |
| Title | City search | | | | |  |  |
| Description | Provide search on particular city | | | | |  |  |
| Actor | User | | | | |  |  |
| Input | Search city | | | | |  |  |
| Behavior | 1.Select the city | | | | |  |  |
|  | 2. view the details of the city | | | | |  |  |
| Output | Particulars of the city selected. | | | | |  |  |
| Precondition | NA | | | | |  |  |
| Post condition | NA | | | | |  |  |
| Exceptions | If there is any problem in retrieving information, display error message. | | | | | | |
| Use case diagram |  |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | User | | | | | Search on city | |



Special instructions NA

13

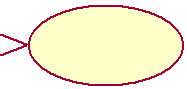
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case2** |  |  |  |  |  |  |  |
| Requirement ID | CS\_UC\_04 | | | | |  |  |
| Title | View city | | | | |  |  |
| Description | It describes the details of the cites, including categories and information | | | | | | |
|  | which user can view | | | | |  |  |
| Actor | User | | | | |  |  |
| Input |  |  |  |  |  |  |  |
| Behavior | 1Search the city | | | | |  |  |
|  | 2.View the city | | | | |  |  |
| Output | Information required by the user | | | | |  |  |
| Precondition | NA | | | | |  |  |
| Post condition | NA | | | | |  |  |
| Exceptions | If there is any problem in Searching information it display error message. | | | | | | |
| Use case diagram |  |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | User | | | | | View city | |
| Special instructions | NA | | | | |  |  |



14

**Test case3**

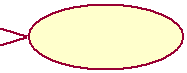
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID | CS\_UC\_05 | |  |  |  |  |  |
| Title | View categories | |  |  |  |  |  |
| Description | It describes the details of the cites including the categories provided by | | | | | |  |
|  | the User. | |  |  |  |  |  |
| Actor | User | |  |  |  |  |  |
| Input | View the categories | |  |  |  |  |  |
| Behavior | 1.Search the city | |  |  |  |  |  |
|  | 2.View the category | |  |  |  |  |  |
| Output | Provide the categories in the city being selected by the user | | | | | |  |
| Precondition | NA | |  |  |  |  |  |
| Post condition | NA | |  |  |  |  |  |
| Exceptions | If there is any problem in retrieving information it displays error message | | | | | |  |
| Use case diagram |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | User | |  |  | View category | |  |
|  |  |  |  |



Special instructions NA

15

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case4** |  |  |  |  |  |  |  |
| Requirement ID | CS\_UC\_04 | |  |  |  |  |  |
| Title | View information | |  |  |  |  |  |
| Description | It describes the information of the category being selected by the User | | | | | |  |
| Actor | User | |  |  |  |  |  |
| Input | View the information | |  |  |  |  |  |
| Behavior | 1.Search the category | |  |  |  |  |  |
|  | 2.View the information | |  |  |  |  |  |
| Output | Provide the information of the category being selected by the user | | | | | |  |
| Precondition | NA | |  |  |  |  |  |
| Post condition | NA | |  |  |  |  |  |
| Exceptions | If there is any problem in retrieving information it displays error message | | | | | |  |
| Use case diagram |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | User | |  |  | View information | |  |
|  |  |  |  |



Special instructions NA

**2.3 Software Requirements Specification**

The goal of requirement specification for a software system - is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as

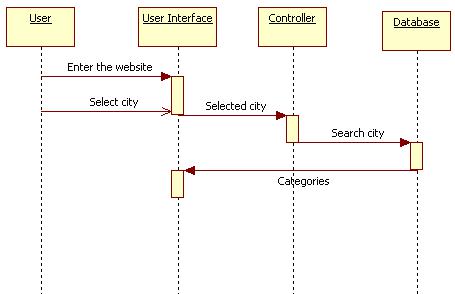
16

functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. On-functional requirements are requirements which impose constraints on the design or implementation

|  |  |
| --- | --- |
| **System feature ID** | CS-SF-01 |
|  |  |
| **Title** | Select the city |
|  |  |
| **Actor** | User |
|  |  |
| **Description** | The purpose of this system feature is to provide selection of city by the user |
|  |  |
| **Input** | Selected city. |
|  |  |
| **Business Logic** | Enter the website. |
|  | Select the city. |
|  | Selected city is forwarded to controller. |
|  | City is searched in the database. |
|  | Result is displayed on the user interface. |
|  |  |
| **Conditions to be checked** |  |
| **on data** | City must be selected. |
|  |  |
| **Output** | Categories available for the selected city. |
|  |  |



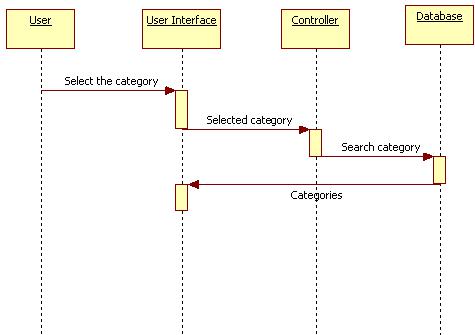
**SEQUENCE DIAGRAM:**



17

|  |  |
| --- | --- |
| **System feature ID** | CS-SF-02 |
|  |  |
| **Title** | Select category |
|  |  |
| **Actor** | User |
|  |  |
| **Description** | The purpose of this system feature is to provide categories available for selected city. |
|  |  |
| **Input** | Selected Category |
|  |  |
| **Business Logic** | Click on categories. |
|  | Select the category. |
|  | Selected category is forwarded to controller. |
|  | Category is searched in the database. |
|  | Result is displayed on the user interface. |
|  |  |
| **Conditions to be checked** |  |
| **on data** | City must be selected |
|  |  |
| **Output** | Displays the categories available for selected city. |
|  |  |

**SEQUENC E DIAGRAM:**

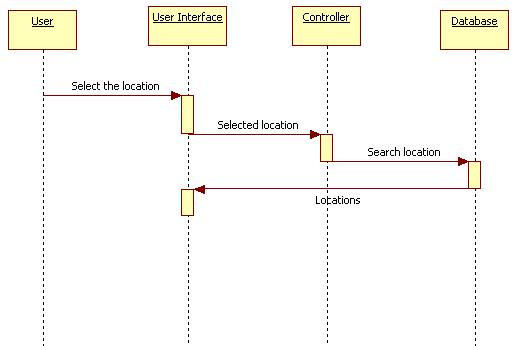


18

|  |  |
| --- | --- |
| **System feature ID** | CS-SF-03 |
|  |  |
| **Title** | Select location |
|  |  |
| **Actor** | User |
|  |  |
| **Description** | The purpose of this system feature is to provide user to select the location |
|  |  |
| **Input** | Selected location |
|  |  |
| **Business Logic** | Click on the location. |
|  | Select the location. |
|  | Selected location is forwarded to controller. |
|  | Location is searched in the database. |
|  | Result is displayed on the user interface. |
|  |  |
| **Conditions to be checked** |  |
| **on data** | Category must be chosen. |
|  |  |
| **Output** | Display the locations available for the category chosen. |
|  |  |



**Sequence Diagram:**

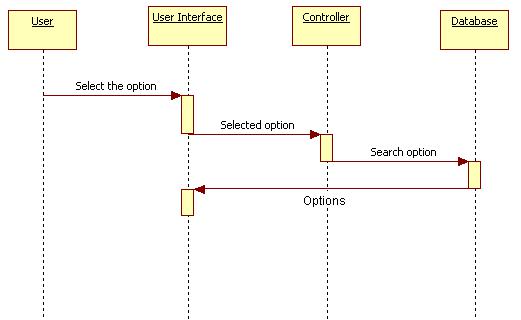


19

|  |  |
| --- | --- |
| **System feature ID** | CS-SF-04 |
|  |  |
| **Title** | Select the option |
|  |  |
| **Actor** | **User** |
|  |  |
| **Description** | The purpose of this system feature is to provide user to select the option |
|  |  |
| **Input** | Selected option |
|  |  |
| **Business Logic** | Click on the option. |
|  | Select the option. |
|  | Selected option is forwarded to controller. |
|  | Option is searched in the database. |
|  | Result is displayed on the user interface. |
|  |  |
| **Conditions to be checked** |  |
| **on data** | Location must be chosen. |
|  |  |
| **Output** | Display the options for selected location. |
|  |  |



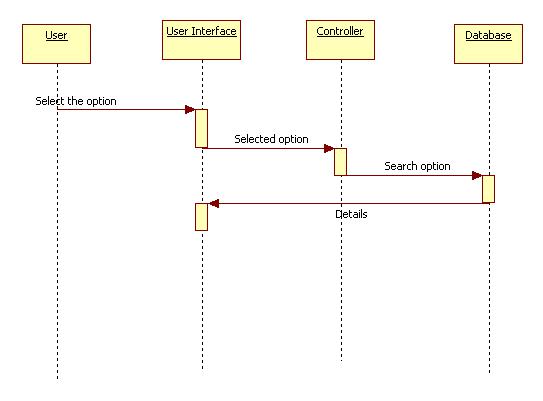
**SEQUENCE DIAGRAM:**



20

|  |  |
| --- | --- |
| **System feature ID** | CS-SF-05 |
|  |  |
| **Title** | Viewing details of the options. |
|  |  |
| **Actor** | User |
|  |  |
| **Description** | The purpose of this system feature is to provide user to select the option |
|  |  |
| **Input** | NA |
|  |  |
| **Business Logic** | Options available. |
|  | View the details. |
|  |  |
| **Conditions to be checked** | Options must be showed. |
| **on data** |  |
|  |  |
| **Output** | Displays the details of the options. |
|  |  |

**SEQUENCE DIAGRAM:**



21

CHAPTER-3

**DESIGN**

**3.0 Introduction**

Software design is a process through which the requirements are translated into a representation of software. The software design involves three technical activities: design, code generation and testing. The design phase is of main importance because in this activity, decisions ultimately affect the success of software implementation and maintenance

**UML DIAGRAM**

UML stands for Unified Modeling Language.UML is a language for specifying, visualizing and documenting the system. This is a step while developing any product after analysis.

**3.1 USECASE DIAGRAM**

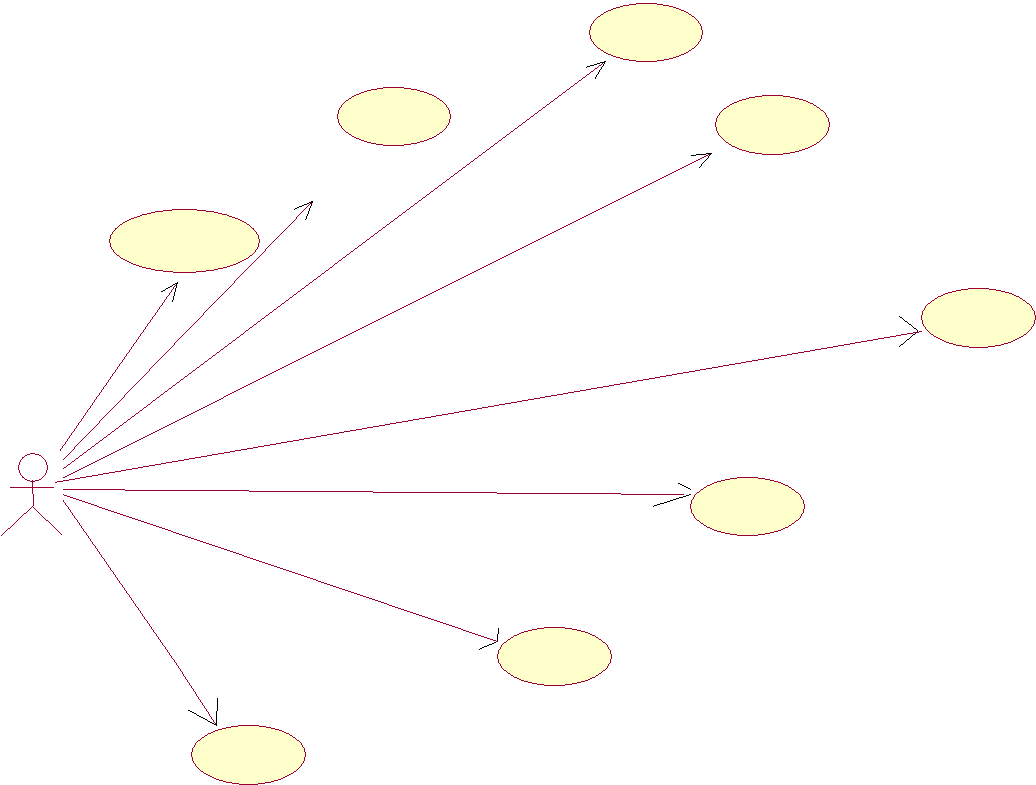
A use case describes an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of use case diagram are use cases and actors.

An actor represents a user or another system that will interact with the system we are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

The use case diagram for the project is as follows:

22

**USECASE DIAGRAM**



**Select**

**category**

|  |  |  |
| --- | --- | --- |
| **View category** | **View location** |  |
|  |  |

**Select city**

**Select location**

|  |  |  |
| --- | --- | --- |
| **User** | **View options** |  |
|  |  |

**Select options**

**View information of desired option**

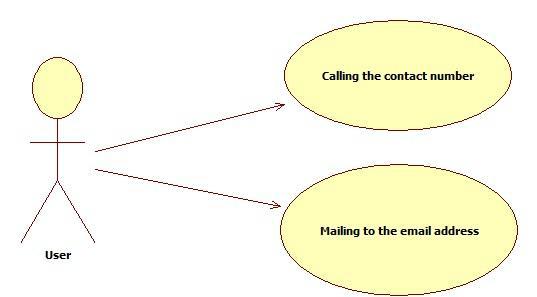


Fig 3.1.1 & 2 Overall use case diagram

23

**3.2 SEQUENCE DIAGRAM**

A sequence diagram is a structured representation of behavior as a series of sequential steps over time. It is used to depict work flow, message passing and how elements in general cooperate over time to achieve a result.

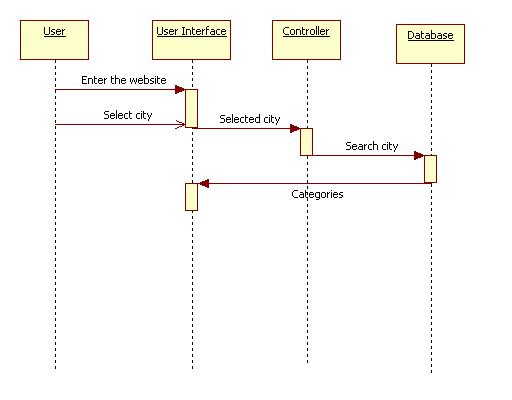


Fig 3.2.1 sequence diagram for user to select the city

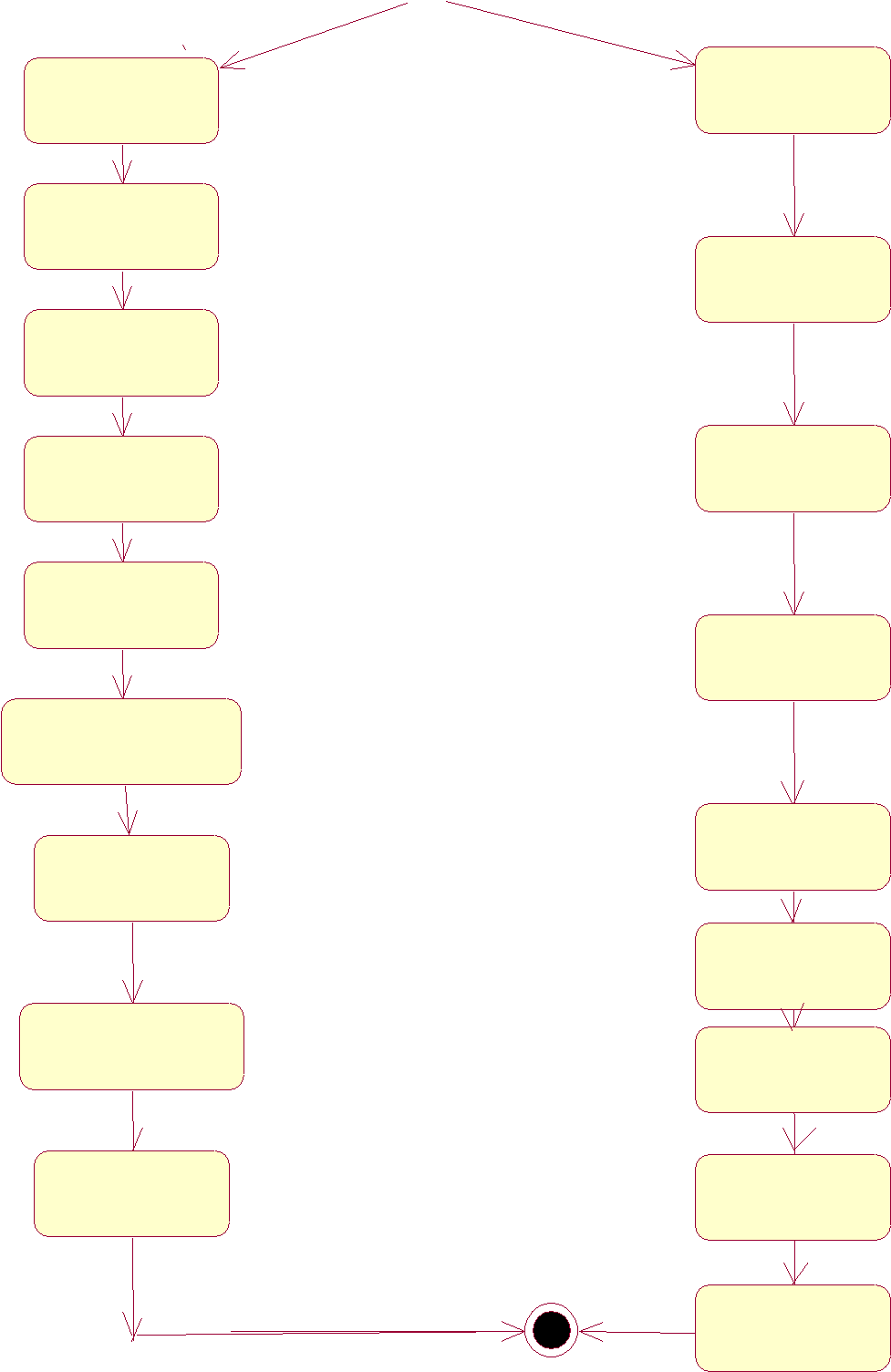
24

**3.3 ACTIVITY DIAGRAM**

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel. Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. The main reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are also useful for: analyzing a use case by describing what actions needs to take place and when they should occur; describing a complicated sequential algorithm; and modeling applications with parallel processes. Activity diagrams do not give detail about how objects behave or how objects collaborate. It displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram focuses on flows driven by internal processing.

25

 Start



Enter website

Select city

Log in

Manage list

Of cities

View category

Select

category

View location

Select location

View option

Select option

Viewinformation Fig 3.4.1 overall activity diagram Of selected

option

Manage

Of category

Manage

List of locations

Inserting

Cities

Inserting

Options

View option

Modify

Options

Logout

26

CHAPTER-4

**IMPLEMENTATION TECHNOLOGY**

**4.0 INTRODUCTION**

System implementation is used to bring a developed system or sub system into operational use and turning it over to the user. It involves programmer users and operational management. Implementation is the process of having systems personnel checkout and put new equipment into use, train users, installs the new application and constructs any files of data needed to use it.

**4.1MODULES AND DESCRIPTION**

**User module:**

**1. Select city**

. Enter the website

. Select the city

. Selected city is forwarded to controller

.City is searched in the database

.Result is displayed on the user interface

**2. Select category**

.Click on category

.Select the category

.Selected category is forwarded to controller

27

.category is searched in the database

.Result is displayed on the user interface

**3. Select location**

.Click on the location

.Select the location

.Selected location is forwarded to controller

.Location is searched in the database

.Result is displayed on the user interface

**4.Select option**

.Click on the option

.Select the option

.Selected option is forwarded to controller

.option is searched in the database

.Result is displayed int the user interface

**5.Viewing details**

.Options available

.View the details

28

**JSP**

JavaServer Pages technology enables you to write standard HTML pages containing tags that run powerful programs based on the Java programming language.JSP technology supports separation of presentation and business logic as follows:



Web designers can design and update pages without learning the Java programming language.



Java technology programmers can write code without having to be concerned with web page design.

**JSP PAGE PROCESSING**



A JSP page must be converted into a servlet in order to service requests.



The web container translates the JSP file into a Java source file that contains a servlet class definition.



The web container compiles the servlet source code into a Java class file.



The servlet class bytecode is loaded into the web container’s JVM software using a classloader.



The web container creates an instance of the servlet class.the web container initializes the servlet by calling the *jspInit* method.



The initialized servlet can now sevice requests.with each request, the web container can call the *\_jspService* method for the converted JSP page.

**HTML**

**HTML**, which stands for **hyper text Markup Language**, is the predominant makeuplanguage for web pages is written in the form of HTML elements consisting of "tags" surrounded by angle brakects within the web page content. It is the building blocks of all basic websites.

29

It allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML webpages.

**CSS**

**Cascading Style Sheets** (**CSS**) is a style sheet language used to describe the presentationsemantics (the look and formatting) of a document written in a markup language. It’s most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document

CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts.[*citation needed*] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for tableless web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. While the author of a document typically links that document to a CSS style sheet, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

30

**JAVA DEVOLPMENT ENVIRONMENT**

To code, edit, debug and test the java programs, one needs to have a java development environment. At the minimum this will consists of a java compiler interpreter and applet viewer where applets can be tested.

Sun’s java development kit (JDK) latest version is 2.2 can be freely downloaded from the Internet.

Java compiler is available on DOS, Win95, WIN’NT, Solaris and MAC etc.

**ABOUT SERVLETS:**

Servlets provides a Java-Based solution used to address the problems currently

associated with doing server side programming, including inextensible scripting

solutions, platform specific APIs, and incomplete interfaces. Servlets are objects conform

to a specific interface that can be plugged into a Java-based server. Servlets are to the

server-side what applets are to the client-side-object byte codes that can be dynamically

loaded off the net. They differ from applets in that they are faceless objects (without

graphics or a GUI component). They serve as platform independent, dynamically

loadable, plug gable helper byte code objects on the server side that can be used to

dynamically extend server-side functionality.

**Configuring the Server:**

The real beauty of the java server is that it is extensible. But, before you can use a Servlet to add an extended functionality to the JavaServer.You have to use the Java Server administration applet to install the Servlet and specify the default parameters and arguments.

31

Display the Administration Applet by connecting to:

[http://server\_Host\_Name:9090/index.html](http://server_host_name:9090/index.html)

Are secure-even when downloading across the network, the servlets security model and the Servlet sandbox to protect your system from unfriendly behavior.

The advantage of the Servlet API

One of the great advantages of the Servlet API is protocol independence. It assumes nothing about:

* The protocol being used to transmit on the net.
* How it is loaded.
* The server environment it will be running in.

**ABOUT ANDROID**

**What is Android?**

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The [Android SDK](http://developer.android.com/sdk/index.html) provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

**Features**

Application framework enabling reuse and replacement of components



Dalvik virtual machine optimized for mobile devices



Integrated browser based on the open source [WebKit](http://webkit.org/) engine



32

Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional) SQLite for structured data storage



Media support for common audio, video, and still image formats (MPEG4, H.264,



MP3, AAC, AMR, JPG, PNG, GIF)

GSM Telephony (hardware dependent)



Bluetooth, EDGE, 3G, and WiFi (hardware dependent)



Camera, GPS, compass, and accelerometer (hardware dependent)



Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE



**Applications**

Android will ship with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language**.**

**Application Framework**

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:

33

A rich and extensible set of [Views](http://developer.android.com/resources/tutorials/views/index.html) that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser [Content Providers](http://developer.android.com/guide/topics/providers/content-providers.html) that enable applications to access data from other applications (such as Contacts), or to share their own data



A [Resource Manager,](http://developer.android.com/guide/topics/resources/resources-i18n.html) providing access to non-code resources such as localized strings, graphics, and layout files



A [Notification Manager](http://developer.android.com/reference/android/app/NotificationManager.html) that enables all applications to display custom alerts in the status bar



An [Activity Manager](http://developer.android.com/reference/android/app/Activity.html) that manages the lifecycle of applications and provides a common navigation backstack



For more details and a walkthrough of an application, see the [Notepad Tutorial.](http://developer.android.com/resources/tutorials/notepad/index.html)

**Android Runtime**

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management**.**

34

**J2EE**

**Java Platform, Enterprise Edition** or **Java EE** is a widely used platform for serverprogramming in the Java programming language. The Java platform (Enterprise Edition) differs from the Java Standard Edition Platform (Java SE) in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

The platform was known as *Java 2 Platform, Enterprise Edition* or *J2EE* until the name was changed to *Java EE* in version 5. The current version is called *Java EE 6*.

Java EE is defined by its specification. As with other Java Community Process specifications, providers must meet certain conformance requirements in order to declare their products as *Java EE compliant*.

Java EE includes several API specifications, such as JDBC, RMI, e-mail, JMS, web services, XML, etc., and defines how to coordinate them. Java EE also features some specifications unique to Java EE for components. These include Enterprise JavaBeans, Connectors, servlets, portlets (following the Java Portlet specification), JavaServer Pages and several web service technologies. This allows developers to create enterprise applications that are portable and scalable, and that integrate with legacy technologies. A Java EE application server can handle transactions, security, scalability, concurrency and management of the components that are deployed to it, in order to enable developers to concentrate more on the business logic of the components rather than on infrastructure and integration tasks.

**ECLIPSE IDE WITH ANDROID PLUGIN**

**Eclipse** is a multi-language software development environment comprising an integrateddevelopment environment (IDE) and an extensible plug-in system. It is written primarily in Java and can be used to develop applications in Java and, by means of various plug-

35

ins, other languages including C, C++, COBOL, Python, Perl, PHP, Scala, Scheme and Ruby (including Ruby on Rails framework). The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C/C++, Eclipse JDT for Java and Eclipse PDT for PHP.

The initial codebase originated from Visual Age. In its default form it is meant for Java developers, consisting of the Java Development Tools (JDT). Users can extend its capabilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. The runtime system of Eclipse is based on Equinox, an OSGi standard compliant implementation.

This plug-in mechanism is a lightweight software componentry framework. In addition to allowing Eclipse to be extended using other programming languages such as C and Python, the plug-in framework allows Eclipse to work with typesetting languages like LaTeX, networking applications such as telnet, and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and CVS support is provided in the Eclipse SDK, with Subversion support provided by third-party plug-ins.

With the exception of a small run-time kernel, everything in Eclipse is a plug-in. This means that every plug-in developed integrates with Eclipse in exactly the same way as other plug-ins; in this respect, all features are "created equal". Eclipse provides plug-ins for a wide variety of features, some of which are through third parties using both free and commercial models.

36

CHAPTER-5

**TESTING AND SCREENS**

**5.0 INTRODUCTION**

Software testing is a critical element of software quality assurance and represents

ultimate reviews of specification, design and coding. The testing phase involves the

testing of the developed system using various test data. While testing the system, errors

are found and corrected using the testing steps and corrections are also noted for future

use.

**5.1 TESTING**

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operations commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct then goal will be successfully achieved. A series of testing is done for the proposed system before system is ready for the user acceptance testing.

**Levels of Testing**

In order to uncover the errors present in different phases we have the concept of testing.

The basic levels of testing are:

1. Unit Testing
2. Integration Testing
3. System Testing

37

**Unit Testing**

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits. Unit tests find problems early in the development cycle. Unit testing allows the programmer to refractor code at a later date, and make sure the module still works correctly (i.e., regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. An elaborate hierarchy of unit tests does not equal integration testing. Integration testing cannot be fully automated and thus still relies heavily on human testers Unit testing provides a sort of living documentation of the system.

**Integration Testing**

Integration testing (sometimes called Integration and Testing.) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using Black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface .Test cases are constructed to test that all components within assemblages

38

interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

**System Testing**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limiting type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement .Specification System testing is an *investigatory* testing phase, where the focus is to have almost a destructive attitude and tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

**Acceptance testing**

Acceptance testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized. Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. User acceptance of a system is the key factor for the success of any system.

39

**White Box Testing**

**White box testing** (a.k.a. clear box testing, glass box testing, transparent box testing, orstructural testing) is a method of testing software that tests internal structures or workings of an application as opposed to its functionality (black box testing). An internal perspective of the system, as well as programming skills, are required and used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While white box testing can be applied at the unit, integration, and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

**Black Box Testing**

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here module will be treated as a black box that will take some input. This generates output. Output for a given set of input combinations are forwarded to other module.

**5.2 TEST RESULTS AND SCREENS**

The user can enter into the website and can click on **search**:

40

**5.3 CONCLUSION**

* The City Search is an online mobile application that provides information about businesses in the various categories available in the city.
* Here the categories involve businesses like Dining, Entertainment, Retails, Travel and Professional Services in the city.
* Various options under these categories will be displayed to the user based on the query.

48

**REFERENCES:**



Java Server Pages Specification. [Online].Available:

<http://java.sun.com/products/jsp/>

NetBeansTM IDE:



<http://wiki.netbeans.org/NetBeansUserFAQ>



Java Plateform,Enterprise Edition Blueprints.[Online].Available:

<http://java.sun.com/j2ee/blueprints/>



Java Servlets Specification.[Online].Available:

<http://java.sun.com/products/servlet/>



CSS Specification

<http://en.wikipedia.org/wiki/CSS>



HTML Specification

<http://en.wikipedia.org/wiki/HTML>



Eclipse IDE with ANDRIOD PLUG-IN

<http://en.wikipedia.org/wiki/Eclipse_software>



J2EE Specification

<http://en.wikipedia.org/wiki/J2EE>

49