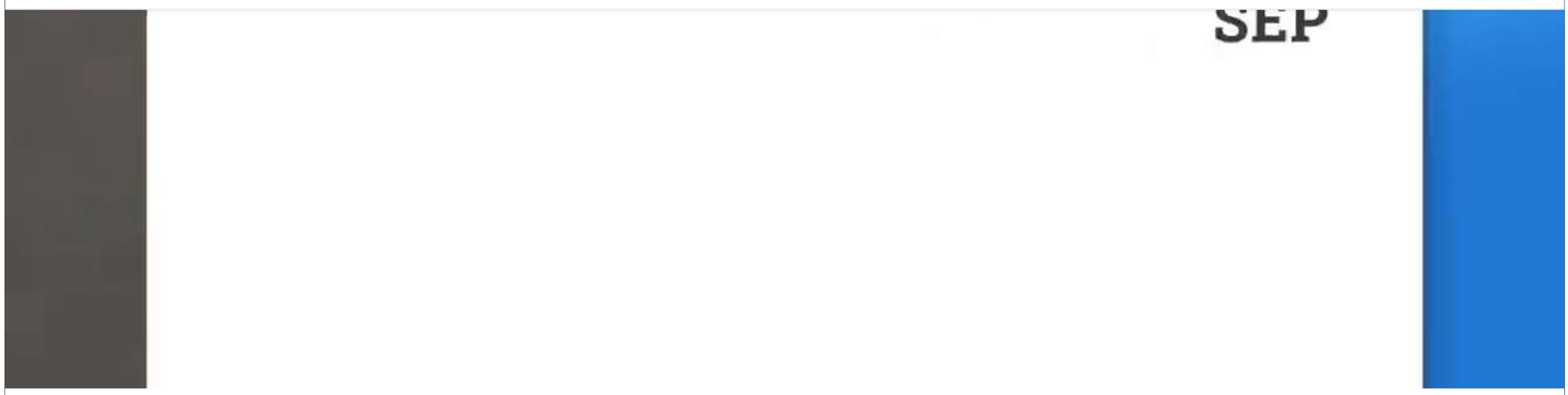
Related titles

SUBMITTED TO **PROF. B.PRAKASH**

SUBMITTED BY:

Index:



00:07 04:00

HD

Related titles

rate on the basis of per day, and the amount to be deposited accordingly.

Cab booking system provides reliable online(web based) cab(car) booking facility to people in various cities in India, free of cost.Cab acts like a bridge between the cab operators and the customers/users/people who book a cab. This is the online cab booking service provided to customers. This brings together the registered travel agencies/cab operators/cab owners and the customers. free service to the travelers/customers/users who go for booking a cab or car or taxi.

2. INTRODUCTION

The purpose of this document is to specify the cab booking functionality.

This kind of development is now present is most of the developing cities .It would be user friendly and makes people on the easier side for transportation in large cities. This would change the traditional system and helpful in time management of the workload in the modern day livelihood.

Here the customers can book a cab /taxi/car by viewing all the cab details and pricing details available, according to selected city and area. It is the reliable service provided to both customers and Travel agencies. This provides service with well conditioned new vehicles, with experienced drivers for a happy journey of the customers. of the 4-wheeler (cars/taxies).

3. OBJECTIVES

- The fare must be economical so that it must be in reach/budget of every person.
- Cab must be reach on time on the defined destination.
- There must be a large fleet of cabs (AC/Non AC).
- Provide the functionality to make your own bookings
- Update your web site without the need to get a web designer involved.
- Provide the customer with taxi availability.
- Track your customers.
- Engage your customers through interaction such as feedback forms
- Easy payment facility must be provided in cab i.e. by cash or by card.
- Payment bill must be provided by cab driver.
- Driver¡¦s identification data must be given i.e. driver¡¦s name, id & photograph at the time of booking of cab.
- Estimated time for a particular journey must be provided.
- Details of the route must be provided to the customer. Customers can my take the cab by his/her own route.
- Customer satisfaction is necessary.
- The user interface must be friendly so that the user can easily book a cab in few minutes by doing few clicks.
- Payment modes can be of prepaid
- If the payment mode is prepaid then the customer have to provide its full name, address, type of card(visa, master, electron-visa etc.), account number, bank name, and branch.
- email id must be provided.
- At the time of booking the web page must have the interface for the starting point, destination, type of cab (AC/Non Ac), charge per kilometre, cab driver details, time, payment options, service area etc.

The information must be provided to the customer on its email id and to driver on its job sheet

4. SOFTWARE REQUIREMENT SPECIFICATION

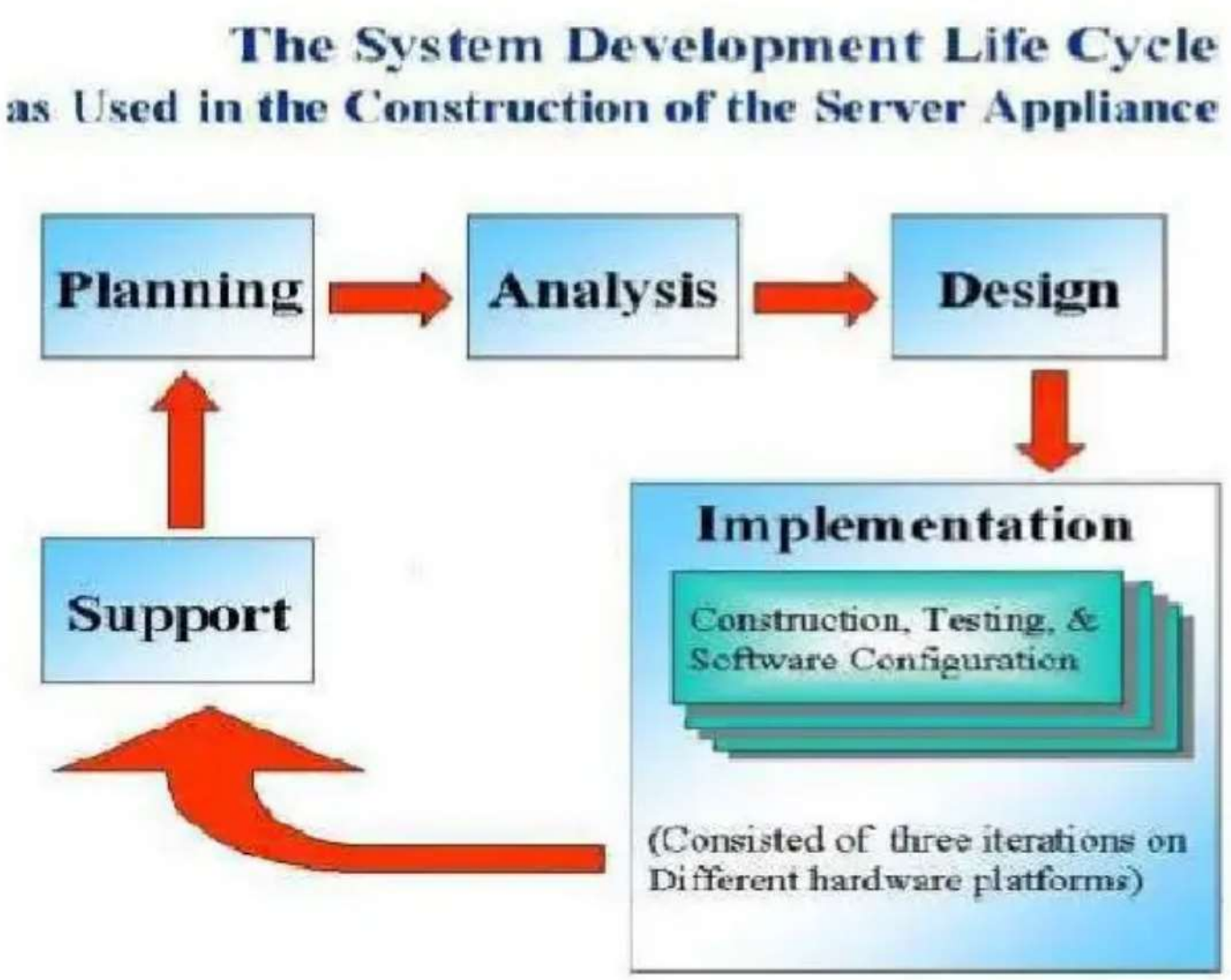
- FULL TAXI DISPATCH SYSTEM web-based. Will include:- - Geolocalisation (Client mode on Dreevo , Iphone, Android, etc...)
- Geolocalization services are provided directly by the satellite to the employees of the company which are provided with the GPS enabled cell phones so that they can track, heck and automize the services of the cabs. - Track via GPRS all the taxis equiped with Dreevo (at least)
- All the cabs are fitted with automated GPRS system and are connected round the clock with the main server for their location information. - SMS function (for dispatch)
- Now all the enquiries offered on the web site www.cabbookings.com are available on your mobile phone through SMS facility. For more information on the mobile service providers and the key words to be used on the mobile, please click here, SMS help .
- Please note that we are giving the backend service only for the SMS queries. For more information and help on key words and SMS facility, kindly contact the mobile service provider according to the table.
- Store in a Database all the customers with ID, Address, Telephone, X, Y.
- The database is maintained with the customer id, address and telephone numbers of all the customers.
- Search the closest available taxis to a specific address of the a Customer (based on a Google Maps Cartography)
- Service provider can search the closest available cab which is nearer to the customers address and the service is based on a technique based on google maps

- Assign a Service to the closest available taxi and change its status to busy, available, soon arrived, off duty, etc....
- When a cab is found then operator must assign its status to busy, available, soon arrived, off duty, etc.... - Keep track of the Service until it finishes and receive approximate distance and time elapsed. - Has a dedicated module to introduce new customers to the database, this is done through address search. - Send the Service via GPRS and receive the information from the taxis via GPRS. (And SMS to !) - Integrated function for payment onboard and invoice system on the web-interface. - The solution must have a customer web-page, that the customer can make its own booking on the site, and can be seen by the operator and the screen with automatic dispatch to available cars and with alert (SMS, SMTP, GRPS...Etc...) - The solution must have an administration tool , for system - The solution must have an Operator Command Panel to lead the dispatch and make the regulation of the taxi traffic, with CHAT and IM in direct with the drivers on the road (through the client : Dreevo, Iphone, Android...)

5. SYSTEM DEVELOPMENT LIFE CYCLE

The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases. Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved. For example, initial project activities might be designated as request, requirements-definition, and planning phases, or initiation, concept-development, and

Related titles



6. PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

Initiation Phase

The Initiation Phase begins when a business sponsor identifies a need or an opportunity. The purpose of the Initiation Phase is to Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need. Identify significant assumptions and conscabts on solutions to that need. Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology, i.e., will a change in the business process offer a solution? Assure executive business and executive technical sponsorship. The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin the project. Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposals purpose, identify expected benefits, and explain how the proposed system supports

Development Phase

The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO. The purpose of the System Concept Development Phase is to: Determine the feasibility and appropriateness of the alternatives. Identify system interfaces. Identify basic functional and data requirements to satisfy the business need. Establish system boundaries identify goals, objectives, critical success factors, and performance measures. Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements. Assess project risks. Identify and initiate risk mitigation actions, and Develop high-level technical architecture, process models, data models, and a concept of operations. This phase explores potential technical solutions within the context of the business need. It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software components, or the decision to use an incremental delivery versus a complete, onetime deployment. Construction of executable prototypes is encouraged to evaluate technology to support the business process.

The System Boundary Document serves as an important reference document to support the Information Technology Project Request (I TPR) process. The I TPR must be approved by the State CIO before the project can move forward.

Requirements Analysis Phase:

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system

performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the

business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and

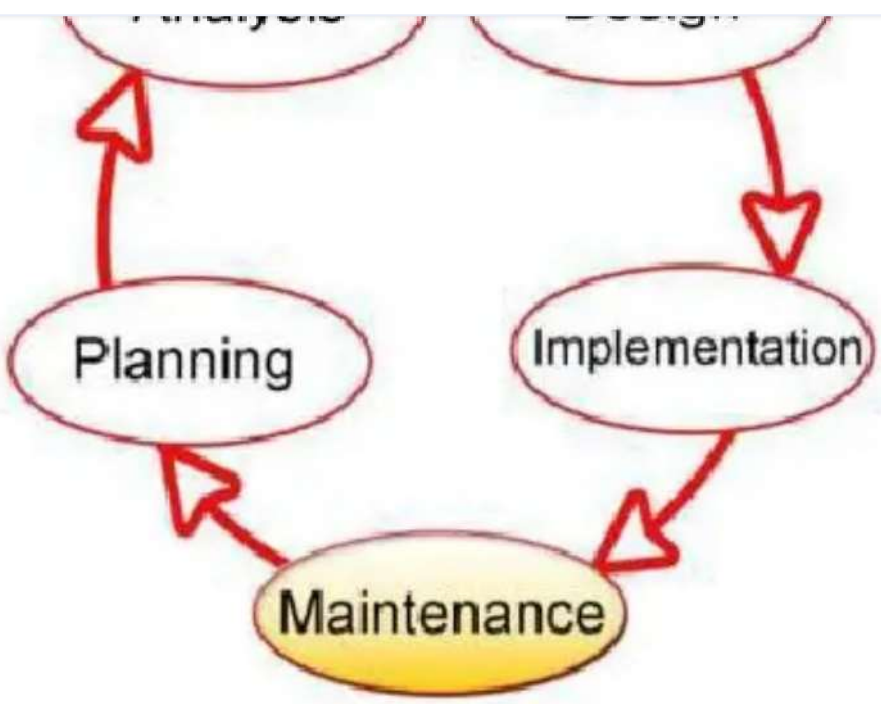
Evaluation Master Plan. The purposes of this phase are to:

Further define and refine the functional and data requirements and document them in the Requirements Document, Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),

Develop detailed data and process models (system inputs, outputs, and the process.

Develop the test and evaluation requirements that will be used to determine acceptable system performance.

Related titles



Design Phase:

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk. These include:

Identifying potential risks and defining mitigating design features. Performing a security risk assessment. Developing a conversion plan to migrate current data to the new system.

Determining the operating environment. Defining major subsystems and their inputs and outputs. Allocating processes to resources. Preparing detailed logic specifications for each software module.

The result is a draft System Design Document which captures the preliminary design for the system. Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system. This document receives a rigorous review by Agency technical and

Related titles

Development Phase:

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction-oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase.

Integration and Test Phase:

Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. Security staff assesses the system security and issue a security certification and accreditation prior to installation/implementation. Multiple levels of testing are performed, including:

Testing at the development facility by the contractor and possibly supported by end users.

Testing as a deployed system with end users working together with contract personnel.

Operational testing by the end user alone performing all functions. Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

Implementation Phase:

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions.

System performance is compared to performance objectives established during the planning phase.

Implementation includes user notification, user cabing, installation of hardware, installation of software onto production until the system is operating in production in accordance with the defined user requirements.

Operations and Maintenance Phase:

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system

Related titles

Certify that the system can process sensitive information.

Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.

Determine when the system needs to be modernized, replaced, or retired.

7. SYSTEM REQUIREMENTS

Minimum system requirements for Cab Management:

- **Processor:** Intel Core 2Duo or AMD equivalent, 2Gz or better
- **RAM:** 2GB
- **HDD:** 160 GB, 7200k spin
- **Operating system:** Windows XP sp3 / Windows Vista Business sp1
- Business broadband connection with static IP (8mb download)
- Online backup
- Logmein, or Windows Remote Desktop Connectivity.

Other suggested components

- 17 inch or larger display
- Secondary hard drive – for additional local backup

Operations:

- Any booking can be done 24 hours.
- One form for single booking.
- Booking is done through pre defined logic.

Communication Interfaces:

The website www.cabbookings.com offeres CBS enquiries on the internet availability, status , fare,service area etc.
Mobile telephone based SMS inquiry service.
Setting up of voice response system.

Product Functions:

It tells the short note about the product.

Cab Details:

Customers may view the cab timing at a date their name and their type of booking.

Booking:

After checking the number of cab available the customers books a cab or number of cabs according to their requiremnets.

Billing:

After reserving the required cab, the customer pays the amount in advance (optional).

User characteristics:

Knowledge user
No voice user
Expert user

Conscabts

Less than 1 sec for local transactions.
3 sec for network transaction.
Uptime of CBS is 99.5+%.

Document Approval

The bill passed on any proposal related to cab management needs approval of the top level management.

8. CODING:

```
import java.awt.*;
import javax.swing.*;

public class UserLoginFrame extends JFrame
{
  JTextField tfuname;
  JPasswordField tfpwd;
  JButton btnlogin;

public UserLoginFrame()
{
   super("Login");
  GridBagConstraints gbc=new GridBagConstraints();

  tfuname= new JTextField(20);
  tfpwd = new JPasswordField(20);
  btnlogin= new JButton("Login");

 Container c=getContentPane();
 c.setLayout(new GridBagLayout());
 setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

  //first label
  c.add(new JLabel("USER NAME:"),gbc);

// user name
gbc.gridwidth=2;
gbc.fill=GridBagConstraints.HORIZONTAL;
gbc.insets=new java.awt.Insets(2,2,2,2);
c.add(tfuname,gbc);

//second label
```

Related titles

```
//password
gbc.gridx=1;
gbc.gridwidth=2;
gbc.insets=new java.awt.Insets(2,2,2,2);
c.add(tfpwd,gbc);

//button
gbc.gridx=0;
gbc.gridy=2;
gbc.gridwidth=3;
gbc.fill= GridBagConstraints.NONE;
gbc.insets=new java.awt.Insets(2,2,2,2);
c.add(btnlogin,gbc);
}
public static void main(String ...args)
{
  UserLoginFrame f=new UserLoginFrame();
  f.setSize(400,300);
  f.setVisible(true);
}
}
```

Final coding:

```
<form>
.
Input elements
.
</form>

First name: <input type="text" name="firstname"><br><br>
Last name: <input type="text" name="lastname"><br><br>
Address:<input type="text" add="address"><br><br>
City:<input type="text" ct="city"<br><br><br>
State:<input type="text" st="state"<br><br><br>
Mobile Number:<input type="number" mb="mobilenumber"<br><br><br>
Time:<input type="number" t="time"<br><br><br>
</form>

Sex :

<input type="radio" name="sex" value="male">Male<br>
<input type="radio" name="sex" value="female">Female<br><br>
</form>

Bording point:<input type="text" bp="boardingpoint"><br><br>
Droping point:<input type="text" dp="dropingpoint"><br><br>
```

```
<input type="radio" car="Car type" value="ac">Ac<br>
<input type="radio" car="Car type" value="non ac">Non Ac<br><br>
</form>
```

Entertainment:

```
<input type="checkbox" name="Entertainment" value="Tv">With Tv<br>
<input type="checkbox" name="Entertainment" value="oTv">With out Tv<br><br>
</form>
```

Proof:

```
<form action="">
<select name="Proof">
<option value="dl">Driving License</option>
<option value="vid">Voters id</option>
<option value="pan" selected>Pancard</option>
<option value="Ac">Adharcard</option>
<option value="psp">Passport</option>
</select>
</form>
```

Fare=Km*hr

```
<form action="">
<input type="button" value="Submit">
</form>
```

9. Test plan

TEST CASES:

1. LOGIN MODULE TESTING

Test case id	1
Test case description	In this stage we are going to test the login part through which the customer logs in through valid user name and password
Module to be tested	Login module
Test data	We are going to give user name and password of the customer as

Related titles

Actual output	Message is displayed
Test result	pass

2. SELECT CAR

Test case id	2
Test case description	In this stage we are going to test selection of car part in which the user is going to select the car he wants
Module to be tested	Select car
Test data	We are going to give the color and company of the car
Expected output	A message must be displayed saying that a car is booked according to the customer requirement
Actual output	Message is displayed
Test result	pass

3. CUSTOMER DETAILS

Test case id	3
Test case description	In this stage we are going to collect the details of the customer and store them.
Module to be tested	Customer details module
Test data	We are going to get the picking and dropping points of the customer and contact details of the customer

Test result	pass

4. CALCULATION MODULE

Test case id	4
Test case description	In this stage we are going to test the calculation part in which the system calculates the amount based on the distance
Module to be tested	Calculation module
Test data	We are going to give the picking and dropping points of the customer as input in customer module
Expected output	Calculated amount must be displayed
Actual output	Amount is displayed
Test result	pass

10. TESTING

Till now the database design, user interface design and implementation are complete. The system now is tested for its functionality, validity and performance. In order to test the system, a wide variety of tests are conducted to make sure that the system matches the entire identified user requirements and conscabts. This chapter focuses on testing the developed systems using different test strategies in order to verify its correctness and user acceptance.
Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

Related titles

activity.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. And it needs to be done in almost every phase of product development life cycle not just before a product is handed to a customer.
The following are some attributes of a good test:
„h A good test has a high probability of finding an error. To achieve this goal the tester must understand the software and attempt to develop a mental picture of how the software may fail. Ideally the classes of failure are probed.
„h A good test is not redundant: testing time and resources are limited. There is no point in conducting the test that has the same purpose as another test. Every test should have a different purpose.
„h A good test should be best of breed. In a group of tests that have a similar intent time and resource limitations may militate for the execution of only a subset of these tests. In such cases the tester that has the highest likelihood of uncovering a whole class of errors should be used.
„h A good test should be neither too simple nor too complex: although it is sometimes possible to combine a series of tests into one test case, the possible side effects associated with this approach may mask errors. In general each test should be executed separately.

Types of testing

White Box Testing:

White box testing, sometimes called glass box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test

Guarantee that all independent paths within a module have been exercised at least once.
Exercise all logical decisions on their true and false sides.
Execute all loops at their boundaries and within their operational bounds.
Exercise internal data structure to assure their validity.
White box testing was performed at all levels of development of i-Admit. The coding team took all care to test the code and guarantee that it meets all the specifications as well as logically correct. All loops were tested and all internal data structures evaluated and verified

It focuses on the functional requirements of the software. That is black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box techniques. Rather it is a complementary approach that is likely to uncover a different class of errors than white box methods. Black box testing attempts to find errors in their following categories

terface errors

Unlike white box testing, which is performed early in the testing process, black box testing is to be applied during later stages of testing.
Black box testing was performed with the application code of the software being developed to verify that it is functionally correct and gives appropriate output at different situations of inputs. It was also verified that

of different types of tests varies as much as the different development approaches.

Unit Testing:

Unit testing focuses verification effort on the smallest unit software design- the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in algorithmic execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths (bases paths) through the control structure are exercised to ensure that all elements in a module have been executed at least once. And finally all error-handling paths are tested.

Application interface of our system was unit tested at all levels of implementation, right from start of code writing, to integrating the code with other modules. Every module was tested fully to check its syntax and logical correctness. Error handling was implemented into relevant modules so that the code doesn't crash on errors.

Integration Testing:

Integration testing is a systematic technique for constructing the program structures, while conducting test to uncover errors associated with interfacing, the objective is to take unit tested modules and build a program structure that has been dictated by design.
User interface of i-Admit was developed in modules. All of them were joined together to make the complete running application. While integrating these modules, integration testing was performed on them to verify that they meet all interfacing requirements and that they pass relevant information among themselves. In the end the complete program structure was tested to ensure interoperability of all the modules.

Validation Testing:

At the culmination of integration testing software is completely assembled as a package: interfacing errors have been uncovered and corrected and a final series of software tests – Validation Testing may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer. Software validation is achieved through a series of Black Box tests that demonstrate conformity with requirements.

Alpha Testing:

It is virtually impossible for a software developer to foresee how the customer will really use a program. When custom software is built for one customer a series of acceptance tests are conducted to enable the customer to validate all requirements.
A customer conducts the alpha test at the developer site. The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problem. Alpha tests are conducted in a controlled environment.
Alpha tests were performed at our development site with the help of our friends, who were called and asked to run the program in the manner they like, without our guidance and errors and usage problems were noted and code was updated to remove all of them.

Beta Testing:

Related titles

Related titles

Related titles

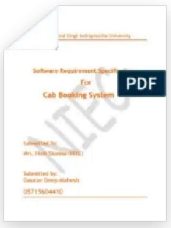
Related titles

Related titles

Related titles

Share this document

You might also like



Document • 44 pages

Cab Booking System

kritikachawla148_993

73% (49)

Document • 31 pages

Car Rental System Project Report

Abhisek Sarkar

83% (6)

Document • 14 pages

Qt Introduction

sunny

No ratings yet

Document • 9 pages

Part One Senior Project

Abdou HK

No ratings yet

Document • 6 pages

Project Proposal

pradeep8968

No ratings yet

Document • 13 pages

Proposal Report Vehicle Parking Management System

PrakashKoirala

40% (5)

Document • 2 pages

Cab Service Project Synopsis

RDGAC CSDepartment

No ratings yet

Document • 29 pages

Related titles

Aman Singh

No ratings yet

Document • 5 pages

final research paper

Aakash singh

No ratings yet

Document • 8 pages

Project Proposal OFFICIAL

brian

No ratings yet

Document • 9 pages

Group-8 car rental system-2

rupa sree

No ratings yet

Show more