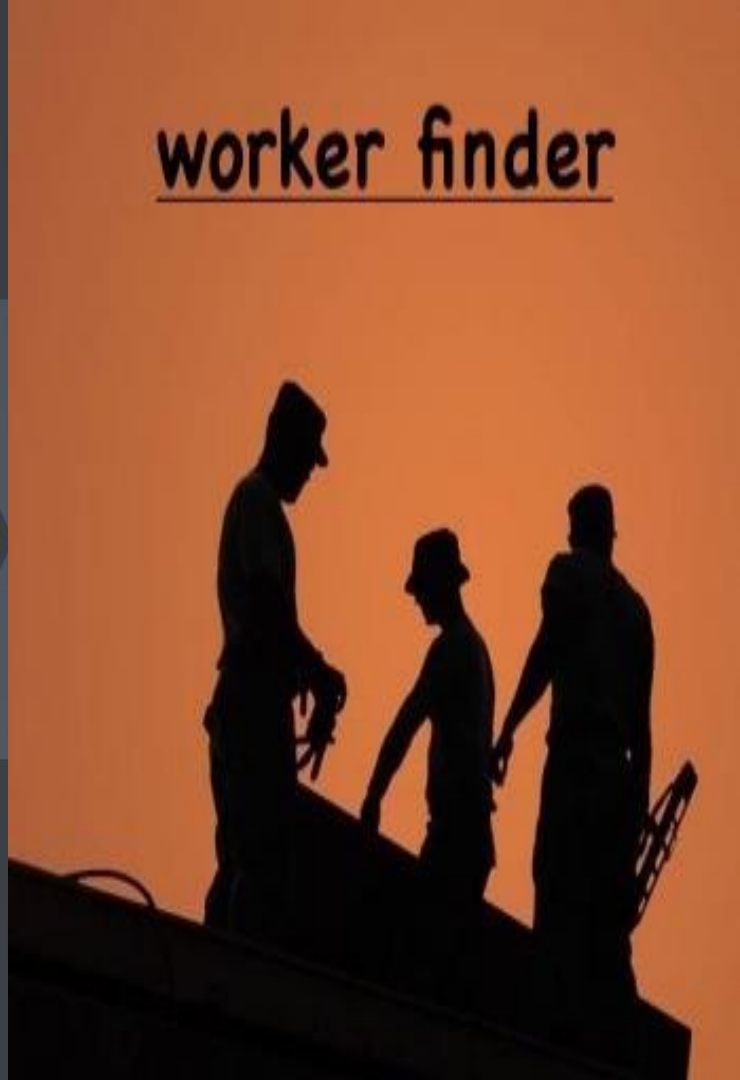

worker finder

GROUP NO:01

MEMBERS NAME

- KULDEEPKUMAR.B.PATEL
  176710316022

INDEX

CHAPTER:1 (INTRODUCTION)

1.1 ABSTRACT

- Application help us to customer to find a worker nearest their area and also help to them book online from the application. It will save time and find a perfect worker as per the needs. Worker can also update their work details and they will earn more money with this concept. Worker can also buy product and repair the customer problem. This Platform help customer and worker to complete the task as per the requirements on tap.

1.2 PROJECT PURPOSE

- Main purpose of the Application is user can find Workers from the Application. Worker can also View the Booking details and after there are this Booking Except/Reject Booking. This App main Review are the multi Field Workers can provide to this Application and then after any Field in Many workers are provided to User. This Application Main Purpose are the Few Time in user Work can Completed in your Time.

1.3 SCOPE

- The platform helps customers hire trusted professionals for services like salon at home, cleaning, plumbing, carpentry, etc.

- **“Worker Finder”** is a app-based marketplace for hiring trusted home and local services. The Company's vision is to use technology and smart processes to structure the highly unorganized services market in India and emerging markets.

- Using **“Worker Finder”** customer can book their services at any time & any place. Customer can see various service categories. Customer can see worker’s details. Customer can see ratings & feedback.

CHAPTER:2 (LITERATURE REVIEW)

- At the current stage in the project, Keeping House aims to address in more depth the key issue of affordability, through a consultative/investigative process with the voluntary sector and groups of older people. To support this process, a small-scale literature review was commissioned from the Institute of Health Sciences and Public Health Research. The brief was to explore the academic and other literature relating to older people's resources, attitudes and expectations in relation to paying for 'keeping house' support.

In particular, the researcher was asked to consider elements which are to be explored within the consultations.

- Older people's attitudes to paying for services
- Incomes and benefits take-up
- Older peoples' expenditure. Do they value housekeeping services?
- Are there particular groups of older people who are disadvantaged?
- Is there evidence about an acceptable price to pay?

The wide range of questions, each with a substantial literature, made the study challenging, especially since there is little empirical work which directly answers the basic question of what will people pay for this service.

Limitation of Existing program:-

- The customers does not get proper workers.
- The customers need to go to various places to get services.
- The customers find difficult to get services they are in need of.
- It is also time consuming.
- Not a better GUI.

CHAPTER:3 (PROJECT MANAGEMENT)

System Requirement Specification

3.1 USER CHARACTERISTIC (MODULE DETAILS) :-

- **ADMIN LOGIN**

- Admin will able to logged into the system using given email and password.
- Admin can manage the whole admin panel and other process of the application.
- Admin can also able to change password and reset password.

- **ADMIN**

- Manage user:
  - Admin can manage the user who has register into app.
- Manage categories:
  - Admin can manage the category like Home Service, Office Service.
- Manage localities:
  - Admin can manage the locations of Ahmedabad city.
- Manage workers:
  - Admin can manage the worker who has register into app and also can verify his personal detail over a call.
  - Admin can also verify the workers.
- Assign workers to the user:
  - Admin can also assign booking if customer not selecting any workers.

- **USER**

- Registration to the app:
  - User can create his account using mobile no and password.
- Login the app:
  - User can login using mobile no and password.
- Search workers according to various categories:
  - User can search various workers on basis of category and area.
  - User can view the workers profile and other detail.
- Book worker by users need:
  - User can select his workers and also can specify the Date and Time for Service booking.
- Feedback:
  - User can also give a feedback to the application.
- Faq:
  - User can view the frequently asked question.

- **WORKER**

- Login into app
  - Worker can create his account and admin will accept / reject account.
- Profile:
  - Worker can update his profile details.
- View booking details:
  - Worker can view booking details.
- Update status:
  - Once user has complete work he can update the status.
  - Worker can view Users locations
  - Workers can view user location for services.

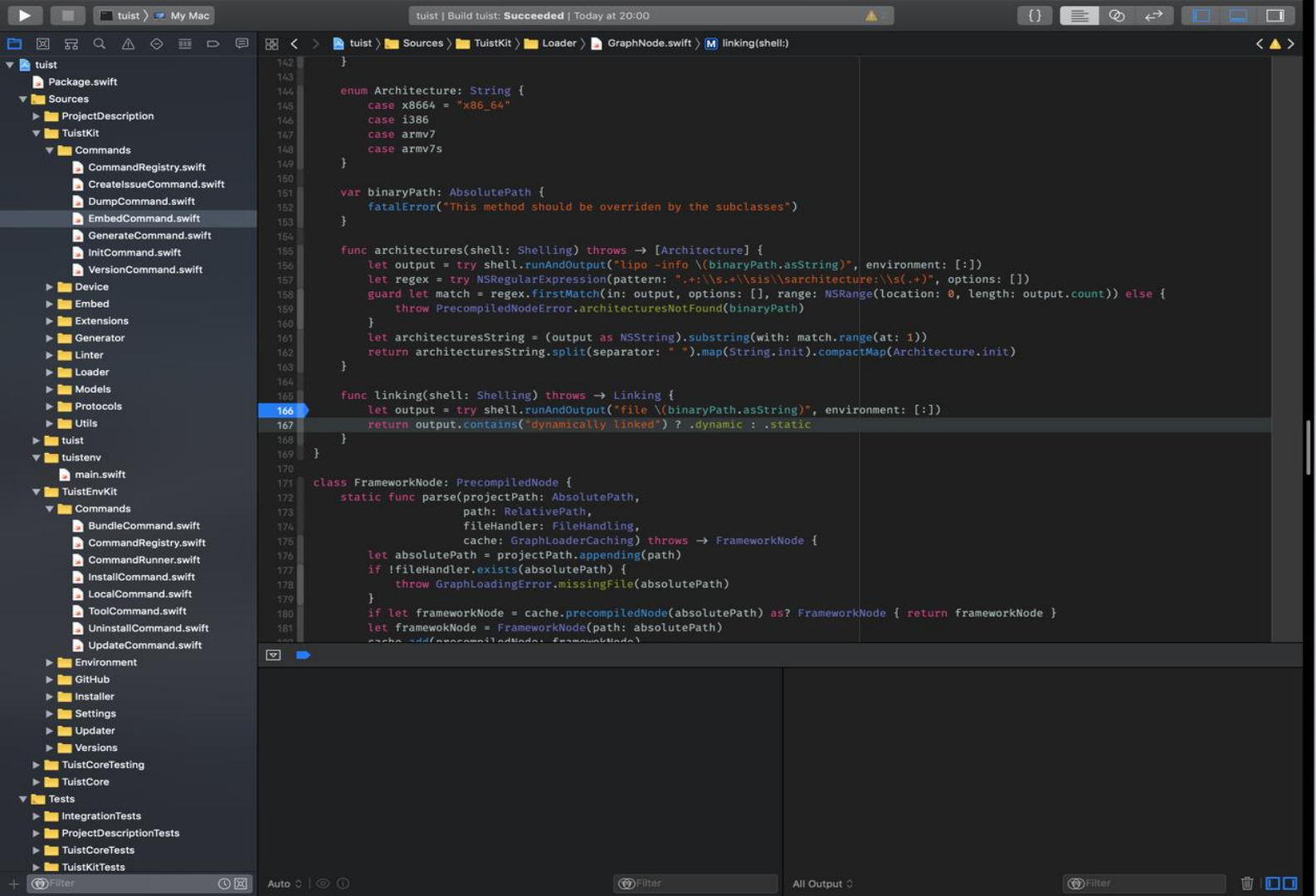
- **PLACEMENT**
- Admin can also assign worker for placement
- Worker can also hire for someone which task will Given by the admin.

- **WORKER COMPER**
- User can compare with another worker in application.

3.2 SOFTWARE REQUIREMENT

Xcode:

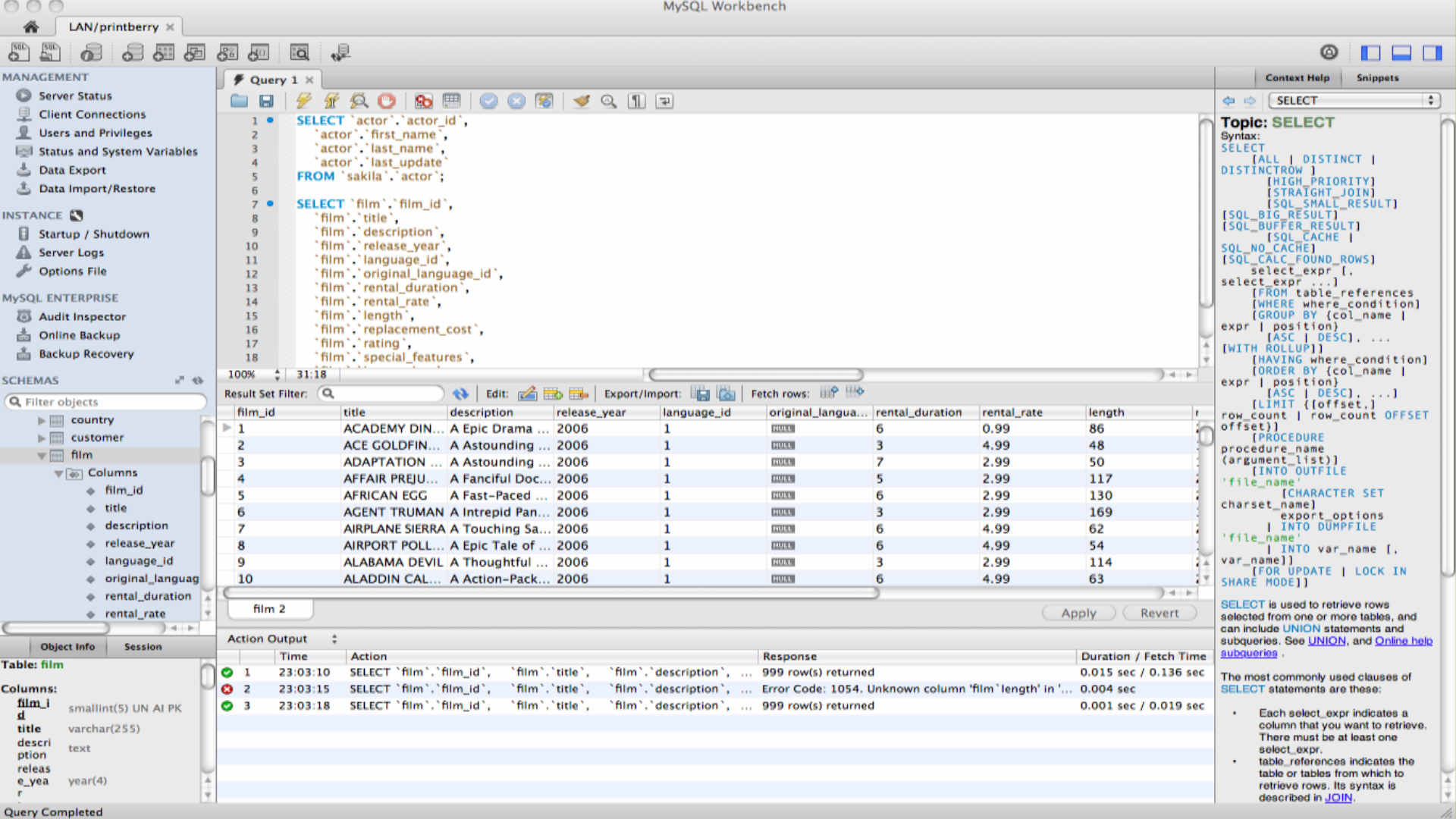
- Xcode is a software package (a set of interrelated programs that work together) used by programmers (actually software engineers and developers) to write software for Mac OS X, iOS devices (iPods, iPhones, iPads), the Apple Watch, and now the Apple TV. Xcode is a type of package called an IDE (Integrated Development Environment) with editors, compilers, and other software tools that work together to help you write software, compile it, load it onto a device, debug it, and ultimately submit it to the app store (or wherever).
- If you write software for one of those devices above, you need Xcode. If you don't write software, well, some of the tools (software programs) are useful, but not generally essential for most people.


tuist | Build tuist: Succeeded | Today at 20:00
tuist 〉 Sources 〉 TuistKit 〉 Loader 〉 GraphNode.swift 〉 linking(shell:)
tuist
Package.swift
Sources
ProjectDescription
TuistKit
Commands
CommandRegistry.swift
CreateIssueCommand.swift
DumpCommand.swift
EmbedCommand.swift
GenerateCommand.swift
InitCommand.swift
VersionCommand.swift
Device
Embed
Extensions
Generator
Linter
Loader
Models
Protocols
Utils
tuist
tuistenv
main.swift
TuistEnvKit
Commands
BundleCommand.swift
CommandRegistry.swift
CommandRunner.swift
InstallCommand.swift
LocalCommand.swift
ToolCommand.swift
UninstallCommand.swift
UpdateCommand.swift
Environment
GitHub
Installer
Settings
Updater
Versions
TuistCoreTesting
TuistCore
Tests
IntegrationTests
ProjectDescriptionTests
TuistCoreTests
TuistKitTests
142 }
143
144 enum Architecture: String {
145 case x8664 = "x86_64"
146 case i386
147 case armv7
148 case armv7s
149 }
150
151 var binaryPath: AbsolutePath {
152 fatalError("This method should be overriden by the subclasses")
153 }
154
155 func architectures(shell: Shelling) throws → [Architecture] {
156 let output = try shell.runAndOutput("lipo -info \(binaryPath.asString)", environment: [:])
157 let regex = try NSRegularExpression(pattern: ".+:\\s.+\\sis\\sarchitecture:\\s(.+)", options: [])
158 guard let match = regex.firstMatch(in: output, options: [], range: NSRange(location: 0, length: output.count)) else {
159 throw PrecompiledNodeError.architecturesNotFound(binaryPath)
160 }
161 let architecturesString = (output as NSString).substring(with: match.range(at: 1))
162 return architecturesString.split(separator: " ").map(String.init).compactMap(Architecture.init)
163 }
164
165 func linking(shell: Shelling) throws → Linking {
166 let output = try shell.runAndOutput("file \(binaryPath.asString)", environment: [:])
167 return output.contains("dynamically linked") ? .dynamic : .static
168 }
169 }
170
171 class FrameworkNode: PrecompiledNode {
172 static func parse(projectPath: AbsolutePath,
173 path: RelativePath,
174 fileHandler: FileHandling,
175 cache: GraphLoaderCaching) throws → FrameworkNode {
176 let absolutePath = projectPath.appending(path)
177 if !fileHandler.exists(absolutePath) {
178 throw GraphLoadingError.missingFile(absolutePath)
179 }
180 if let frameworkNode = cache.precompiledNode(absolutePath) as? FrameworkNode { return frameworkNode }
181 let framewokNode = FrameworkNode(path: absolutePath)
Auto
All Output
8:27
Watch
Files
Extras

DATABASE

- **MYSQL: -** MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.
- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.


MySQL Workbench
LAN/printberry
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
MySQL ENTERPRISE
Audit Inspector
Online Backup
Backup Recovery
SCHEMAS
Filter objects
country
customer
film
Columns
film_id
title
description
release_year
language_id
original_languag
rental_duration
rental_rate
Object Info
Session
Table: film
Columns:
film_id smallint(5) UN AI PK
title varchar(255)
description text
release_year year(4)
Query 1
SELECT `actor`.`actor_id`,
`actor`.`first_name`,
`actor`.`last_name`,
`actor`.`last_update`
FROM `sakila`.`actor`;
SELECT `film`.`film_id`,
`film`.`title`,
`film`.`description`,
`film`.`release_year`,
`film`.`language_id`,
`film`.`original_language_id`,
`film`.`rental_duration`,
`film`.`rental_rate`,
`film`.`length`,
`film`.`replacement_cost`,
`film`.`rating`,
`film`.`special_features`,
100%
31:18
Result Set Filter:
Edit:
Export/Import:
Fetch rows:
film_id | title | description | release_year | language_id | original_langua... | rental_duration | rental_rate | length
1 | ACADEMY DIN... | A Epic Drama ... | 2006 | 1 | NULL | 6 | 0.99 | 86
2 | ACE GOLDFIN... | A Astounding ... | 2006 | 1 | NULL | 3 | 4.99 | 48
3 | ADAPTATION ... | A Astounding ... | 2006 | 1 | NULL | 7 | 2.99 | 50
4 | AFFAIR PREJU... | A Fanciful Doc... | 2006 | 1 | NULL | 5 | 2.99 | 117
5 | AFRICAN EGG | A Fast-Paced ... | 2006 | 1 | NULL | 6 | 2.99 | 130
6 | AGENT TRUMAN | A Intrepid Pan... | 2006 | 1 | NULL | 3 | 2.99 | 169
7 | AIRPLANE SIERRA | A Touching Sa... | 2006 | 1 | NULL | 6 | 4.99 | 62
8 | AIRPORT POLL... | A Epic Tale of ... | 2006 | 1 | NULL | 6 | 4.99 | 54
9 | ALABAMA DEVIL | A Thoughtful ... | 2006 | 1 | NULL | 3 | 2.99 | 114
10 | ALADDIN CAL... | A Action-Pack... | 2006 | 1 | NULL | 6 | 4.99 | 63
film 2
Apply
Revert
Action Output
Time | Action | Response | Duration / Fetch Time
1 | 23:03:10 | SELECT `film`.`film_id`, `film`.`title`, `film`.`description`, ... | 999 row(s) returned | 0.015 sec / 0.136 sec
2 | 23:03:15 | SELECT `film`.`film_id`, `film`.`title`, `film`.`description`, ... | Error Code: 1054. Unknown column 'film`length' in '... | 0.004 sec
3 | 23:03:18 | SELECT `film`.`film_id`, `film`.`title`, `film`.`description`, ... | 999 row(s) returned | 0.001 sec / 0.019 sec
Query Completed
Context Help
Snippets
SELECT
Topic: SELECT
Syntax:
SELECT
[ALL | DISTINCT | DISTINCTROW]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT]
[SQL_BIG_RESULT]
[SQL_BUFFER_RESULT]
[SQL_CACHE | SQL_NO_CACHE]
[SQL_CALC_FOUND_ROWS]
select_expr [, select_expr ...]
[FROM table_references
[WHERE where_condition]
[GROUP BY {col_name | expr | position}
[ASC | DESC], ... [WITH ROLLUP]]
[HAVING where_condition]
[ORDER BY {col_name | expr | position}
[ASC | DESC], ...]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[PROCEDURE procedure_name (argument_list)]
[INTO OUTFILE 'file_name'
[CHARACTER SET charset_name]
export_options
| INTO DUMPFILE 'file_name'
| INTO var_name [, var_name]]
[FOR UPDATE | LOCK IN SHARE MODE]]
SELECT is used to retrieve rows selected from one or more tables, and can include UNION statements and subqueries. See UNION, and Online help subqueries .
The most commonly used clauses of SELECT statements are these:
Each select_expr indicates a column that you want to retrieve. There must be at least one select_expr.
table_references indicates the table or tables from which to retrieve rows. Its syntax is described in JOIN.

3.3 SOFTWARE REQUIRMENT

Language	iOS With Swift
IDE	Xcode
Platform	Mac OS
Database	MySQL
Documentation Tools	Word

3.4 HARDWARE REQUIREMENT

Hard Disk	16GB
RAM	4GB
Processor	intel Core i5

3.5 PROJECT DEVELOPMENT APPROACH

- Software development life cycle **(SDLC)** is a series of phase that provide a common understanding of the software building process. Traditionally, the Systems Development Lifecycle (SDLC) and Project Management (PM) methodologies have followed a waterfall or gated approach. Depending upon the project and project organization, the methodologies may be adapted using waterfall, iterative, incremental, spiral or agile approaches. In our project we are using Spiral model for showing our project development approach.
- The spiral model is similar to the incremental model, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation.
- A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral. It‘s one of the software development models like Waterfall, Agile, and V-Model.

Planning Phase :-

Requirements are gathered during the planning phase. Requirements like BRS that is Business Requirement Specifications and SRS that is System Requirement specifications.

Risk Analysis :-

In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. If any risk is found during the risk analysis then alternate solutions are suggested and implemented.

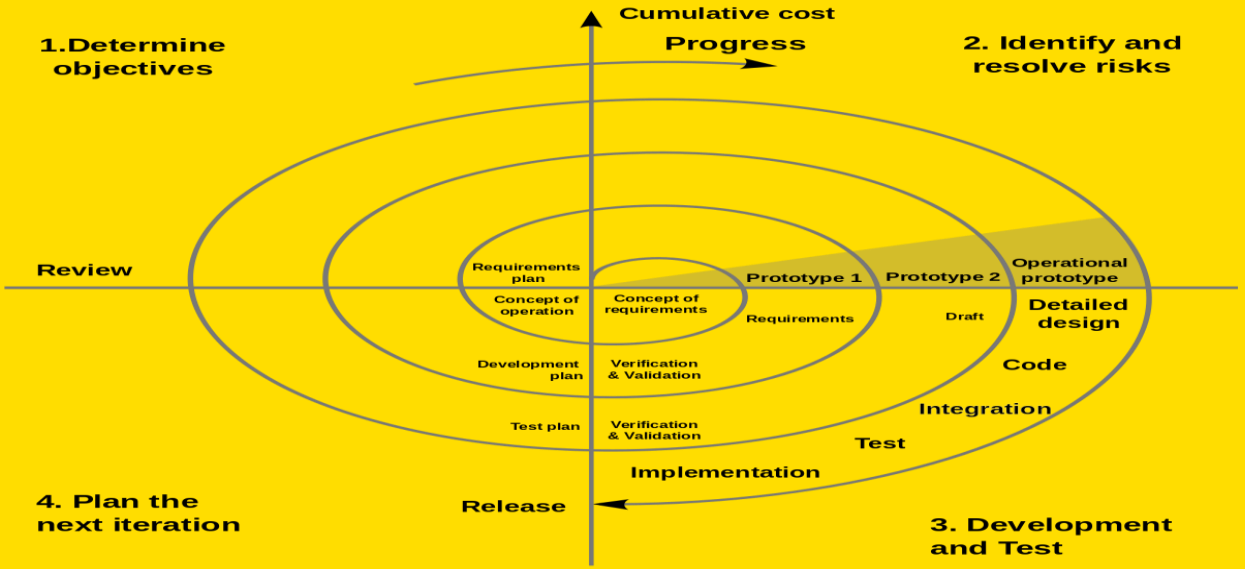
Engineering Phase :-

In this phase software is developed, along with testing at the end of the phase. Hence in this phase the development and testing is done.

Evaluation phase:

This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

SPIRAL MODEL DIAGRAM



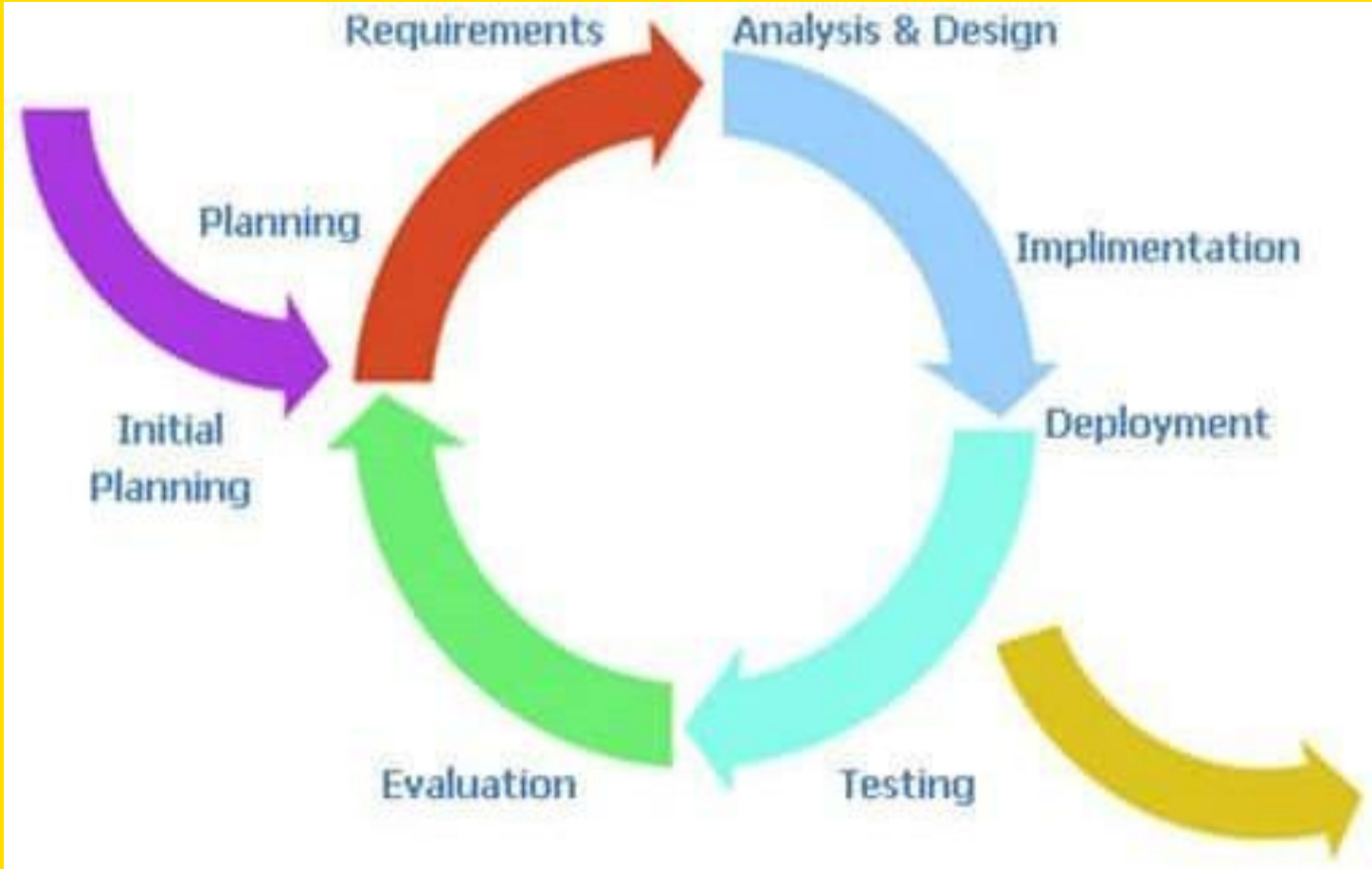
ADVANTAGES OF SPIRAL MODEL

- High amount of risk analysis hence, avoidance of Risk is enhanced.
- Good for large and mission-critical projects.
- Strong approval and documentation control.
- Additional Functionality can be added at a later date.
- Software is produced early in the software life cycle.

DISADVANTAGES OF SPIRAL MODEl

- Can be a costly model to use.
- Risk analysis requires highly specific expertise.
- Project‘s success is highly dependent on the risk analysis phase.
- Doesn‘t work well for smaller projects.

WHEN TO USE SPIRAL MODEL



3.6 SCHEDULE REPRESENTSTION

Week	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
Month / Activity	June		July				August				September				October		
Deciding project title& platform																	
Requirement gathering																	
Deciding development approach																	
Data modelling																	
Design																	
Preparing report																	

CHAPTER :4 (ANALYSIS & DESIGN)

4.1 Feasibility Study

4.1.1 Technical Feasibility:-

Technical Feasibility refers to the ability of the process to take advantage of the current state of the technology in pursuing further improvement.

In this type of study the current technology in used in an organization is checked such as the existing software, hardware, and personnel staff to determine whether it will work for the proposed system or completely new ones is to be used.

The technology that was important in developing a new system such as Development tools, back-end database system were available from within the organization.

4.1.2 Economical Feasibility:-

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study.

Generally, it means whether a business or a project is feasible cost wise and logistically. Economists calculate economic feasibility by analysing the costs and revenues a business would incur by undertaking a certain project.

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study.

These include:

- Economic feasibility cash flow.
- Estimated total project cost.
- Estimated total earnings.
- Risk factors.
- Cost benefits.

4.1.3 Optional Feasibility:-

- **Operational feasibility** refers to the measure of solving problems with the help of a new proposed system. It helps in taking advantage of the opportunities and fulfils the requirements as identified during the development of the project. It takes care that the management and the users support the project.
- After analyzing the technical, economic, and scheduling feasibility studies, next would come the operational analysis. In order to determine if the redesign of the workspace environment would work, an example of an operational feasibility study would follow this path based on six elements:
- **Process** – Input and analysis from everyone the new redesign will affect along with a data matrix on ideas and suggestions from the original plans.

- **Evaluation** –Determinations from the process suggestions; will the redesign benefit everyone? Who is left behind? Who feels threatened?
- **Implementation** –Identify resources both inside and out that will work on the redesign. How will the redesign construction interfere with current work?
- **Resistance** –What areas and individuals will be most resistant? Develop a change resistance plan.
- **Strategies** – How will the organization deal with the changed or structures need to be reviewed or implemented in order for the redesign to be effective?
- **Adapt & Review** –How much time does the organization need to adapt to the new redesign? How will it be reviewed and monitored? What will happen if, through a monitoring process, additional changes must be made?

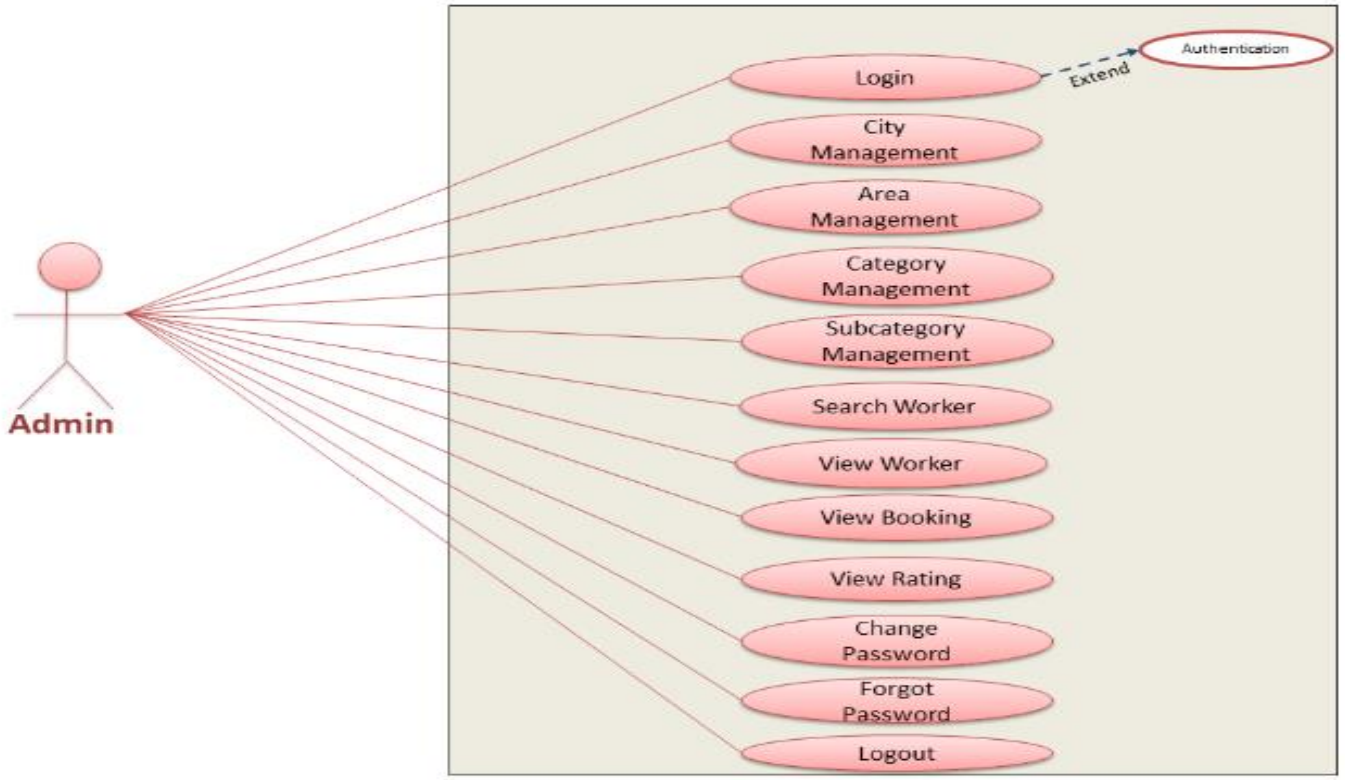
A) RISK ANALYSIS:-

- Risk analysis is the review of the risks associated with a particular event or action. It is applied to projects, information technology, security issues and any action where risks may be analysed on a quantitative and qualitative basis. Risk analysis is a component of risk management.
- Risks are part of every project and business endeavours. As such, risk analysis should occur on a recurring basis and be updated to accommodate new potential threats. Strategic risk analysis minimizes future risk probability and damage
- The risk management process involves a few key steps. First, potential threats are identified. For example, risks are associated with individuals using a computer either incorrectly or inappropriately, which creates security risks. Risks are also related to projects that are not completed in a timely manner, resulting in significant costs.

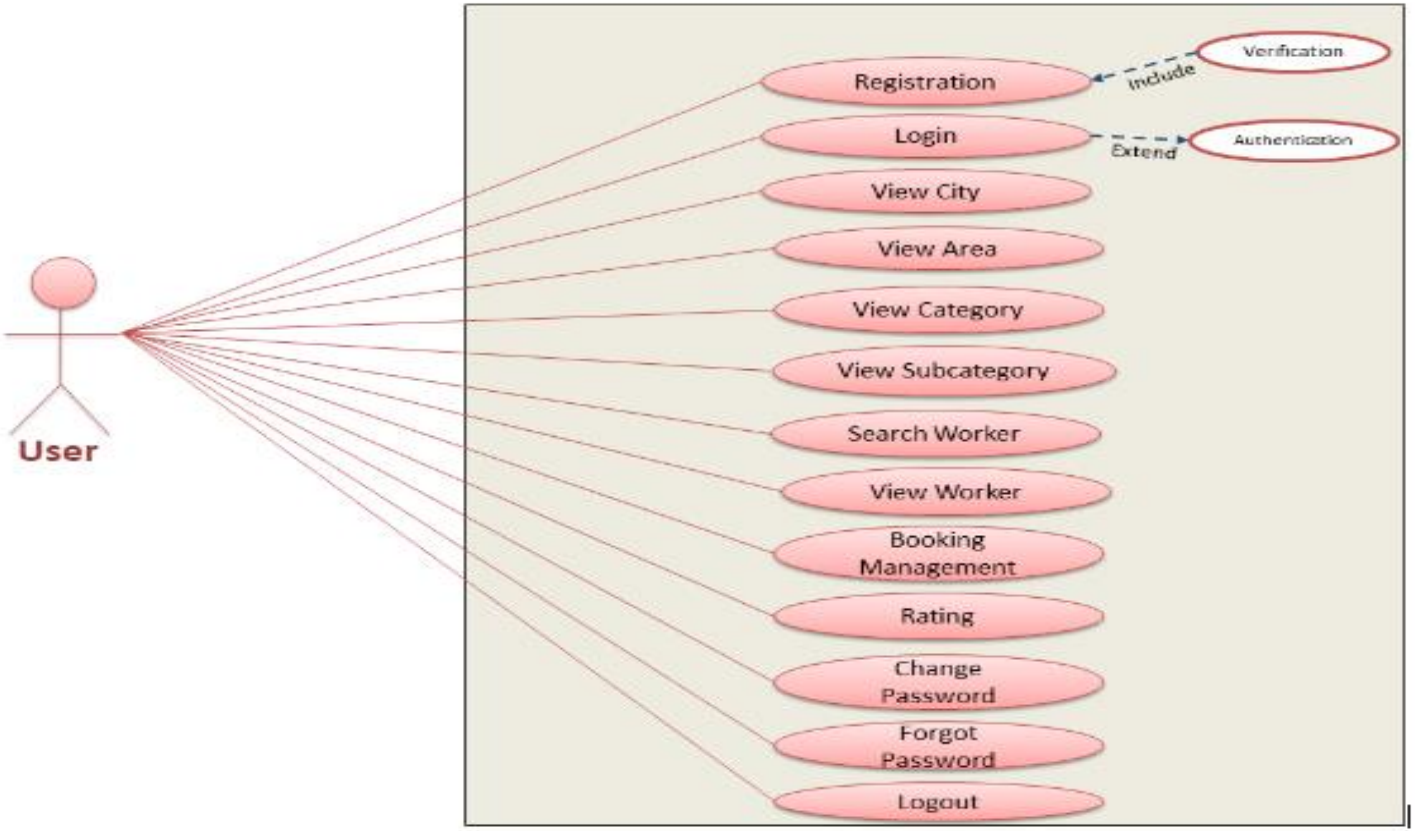
- Next, quantitative and/or qualitative risk analysis is applied to study identified risks. Quantitative risk analysis measures expected risk probability to forecast estimated financial losses from potential risks. Qualitative risk analysis does not use numbers but reviews threats and determines and establishes risk mitigation methods and solutions.
- A contingency plan may be used during risk analysis. If a risk is presented, contingency plans help minimize damage.

4.2 Function of System:-

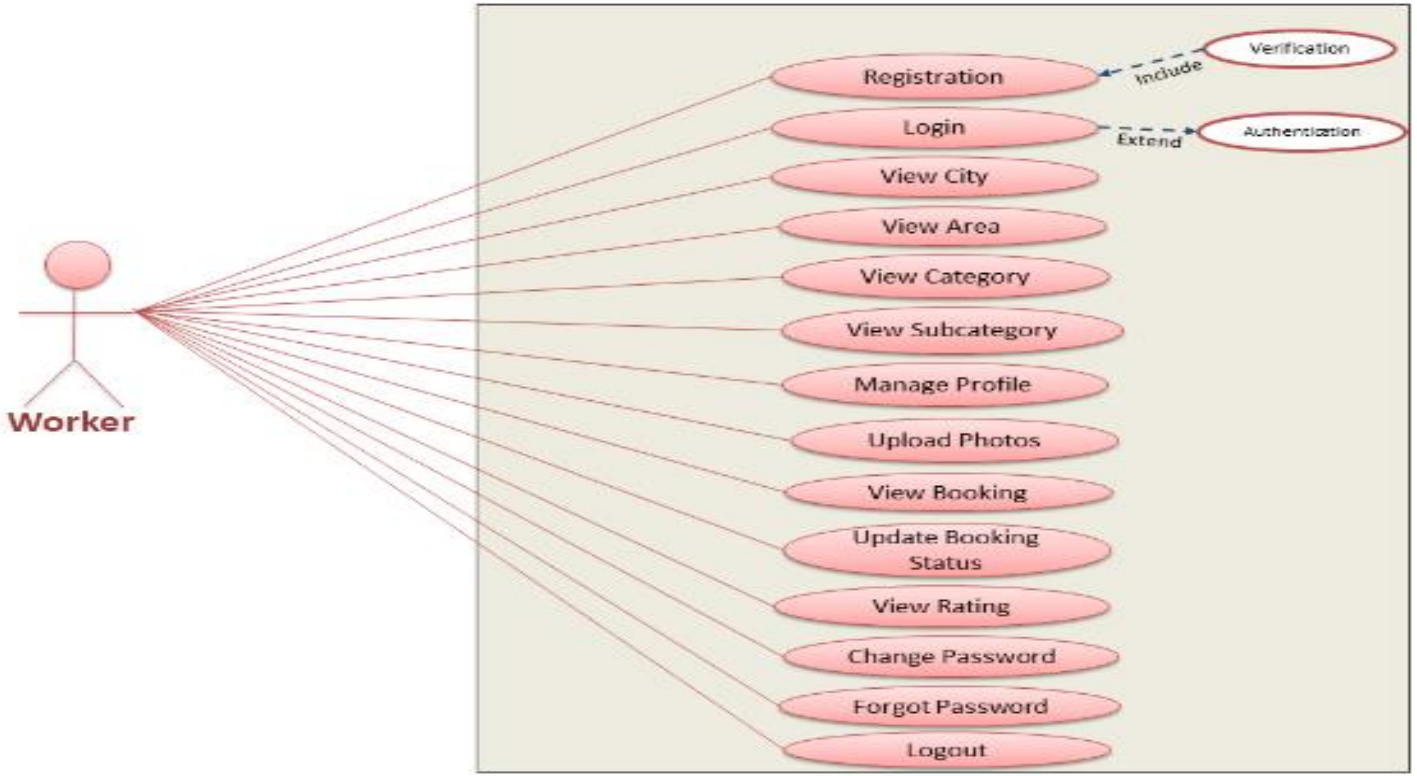
4.2.1 Use Case Diagram For Admin:-


Admin
Login
Extend
Authentication
City Management
Area Management
Category Management
Subcategory Management
Search Worker
View Worker
View Booking
View Rating
Change Password
Forgot Password
Logout

Use Case Diagram For User:-

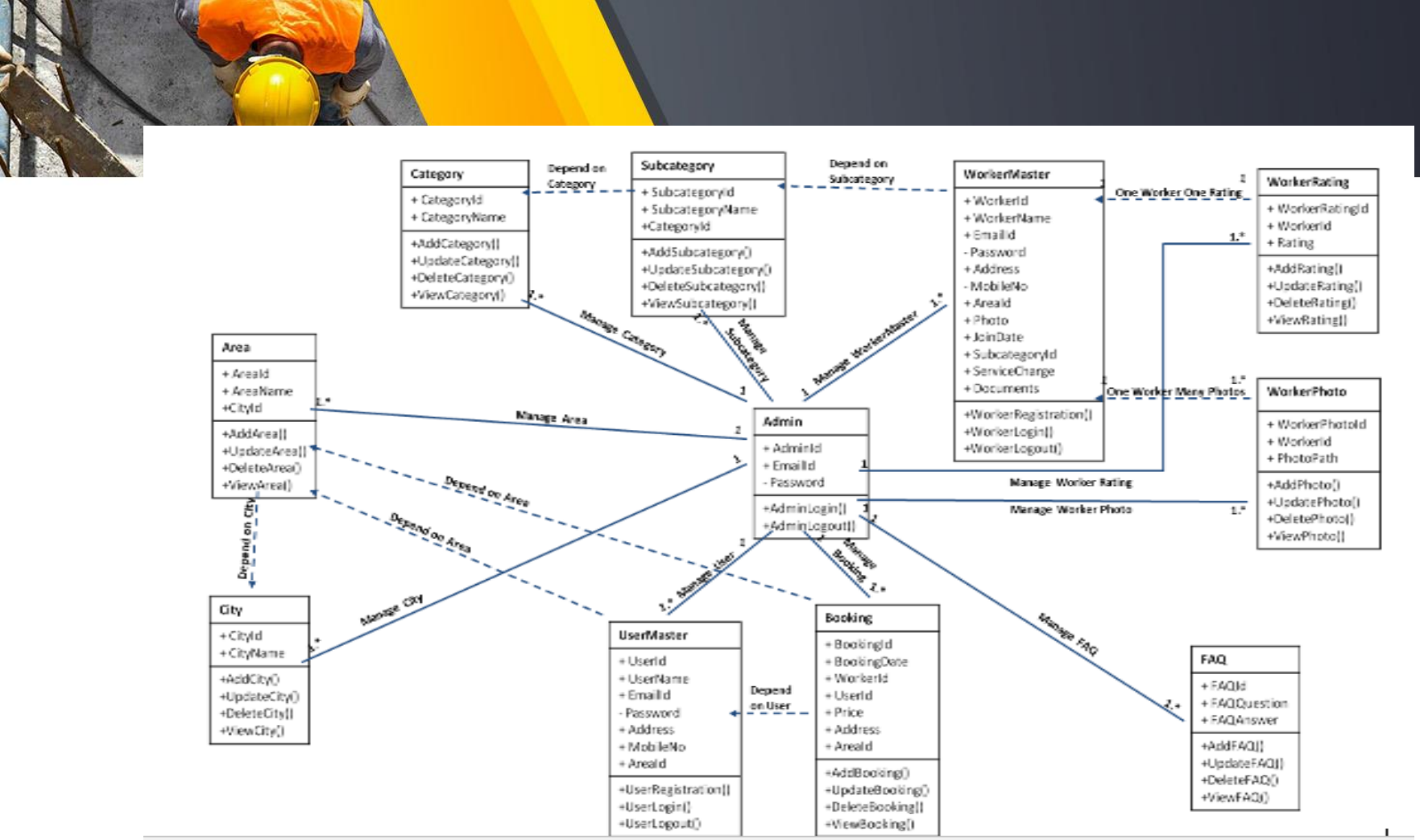


Use Case Diagram For Worker:-

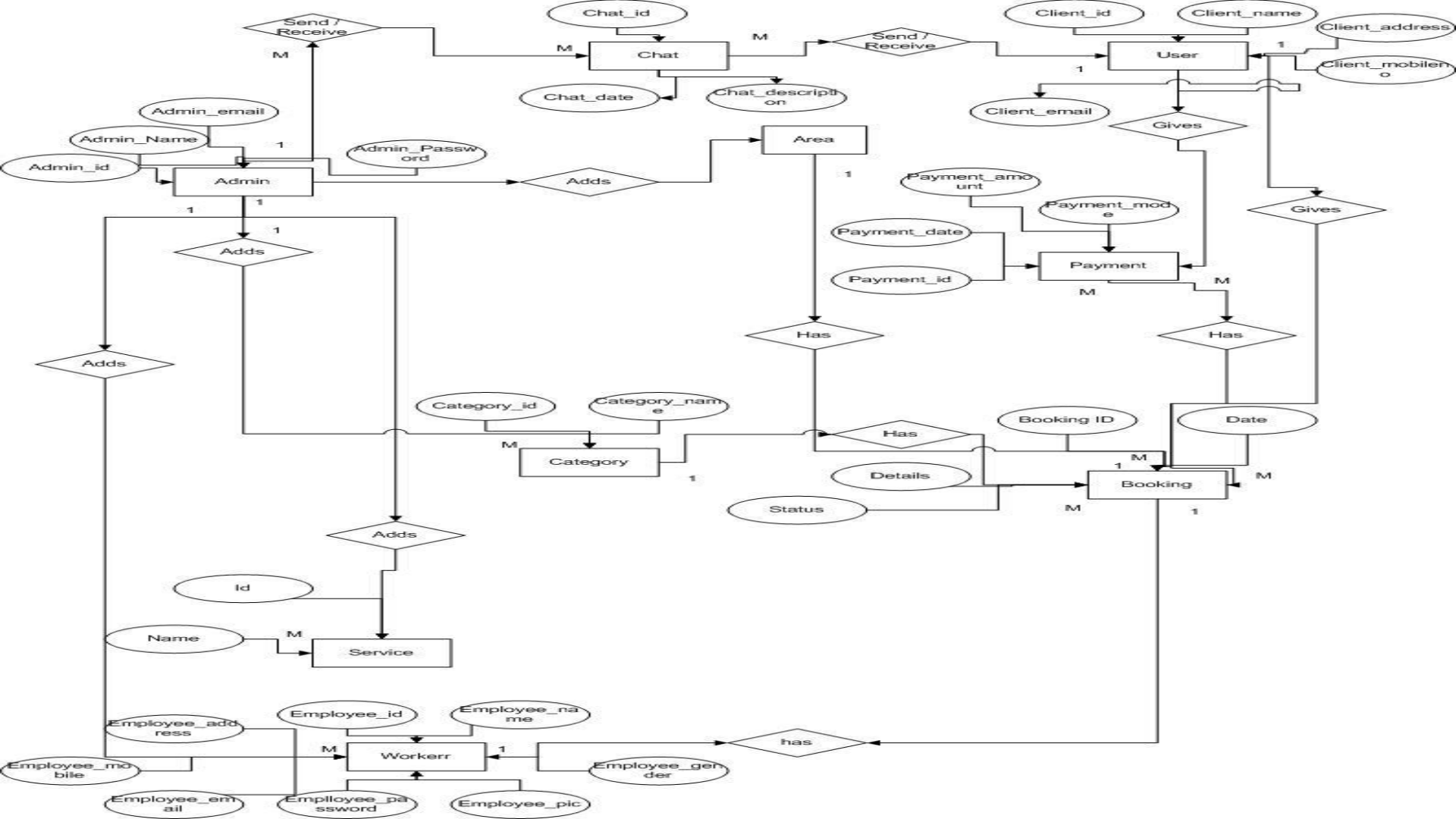


4.3 Data Modeling:-

4.3.1 Class Diagram:-

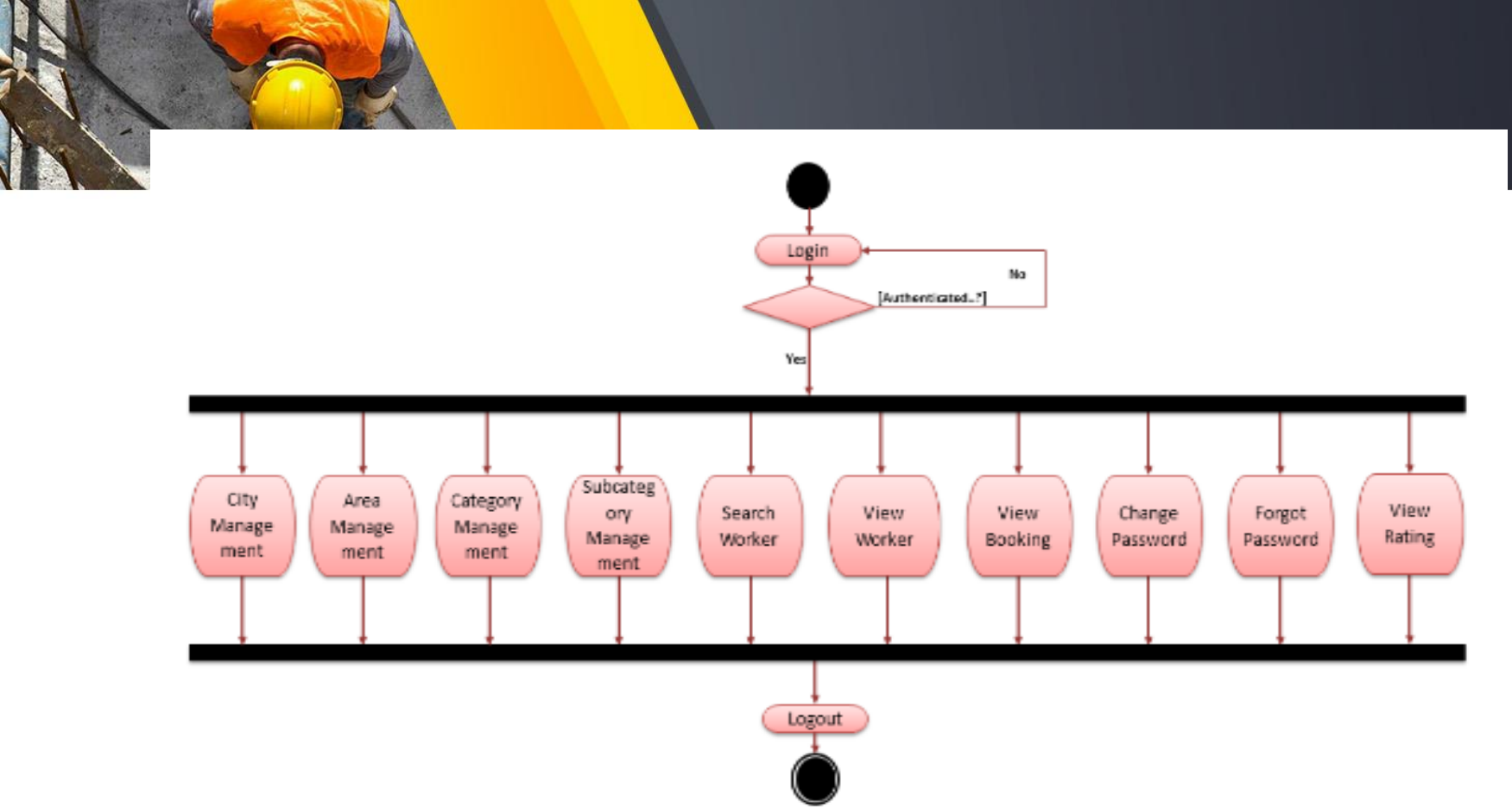

Category
+ CategoryId
+ CategoryName
+AddCategory()
+UpdateCategory()
+DeleteCategory()
+ViewCategory()
Depend on Category
Subcategory
+ SubcategoryId
+ SubcategoryName
+CategoryId
+AddSubcategory()
+UpdateSubcategory()
+DeleteSubcategory()
+ViewSubcategory()
Depend on Subcategory
WorkerMaster
+WorkerId
+WorkerName
+EmailId
- Password
+Address
- MobileNo
+AreaId
+Photo
+JoinDate
+SubcategoryId
+ServiceCharge
+Documents
+WorkerRegistration()
+WorkerLogin()
+WorkerLogout()
One Worker One Rating
WorkerRating
+ WorkerRatingId
+ WorkerId
+ Rating
+AddRating()
+UpdateRating()
+DeleteRating()
+ViewRating()
One Worker Many Photos
WorkerPhoto
+ WorkerPhotoId
+ WorkerId
+ PhotoPath
+AddPhoto()
+UpdatePhoto()
+DeletePhoto()
+ViewPhoto()
Manage Category
Manage Subcategory
Manage WorkerMaster
Area
+ AreaId
+ AreaName
+CityId
+AddArea()
+UpdateArea()
+DeleteArea()
+ViewArea()
Manage Area
Admin
+ AdminId
+ EmailId
- Password
+AdminLogin()
+AdminLogout()
Manage Worker Rating
Manage Worker Photo
Depend on City
Depend on Area
Depend on Area
City
+ CityId
+ CityName
+AddCity()
+UpdateCity()
+DeleteCity()
+ViewCity()
Manage City
Manage User
Manage Booking
UserMaster
+ UserId
+ UserName
+ EmailId
- Password
+ Address
+ MobileNo
+ AreaId
+UserRegistration()
+UserLogin()
+UserLogout()
Depend on User
Booking
+ BookingId
+ BookingDate
+ WorkerId
+ UserId
+ Price
+ Address
+ AreaId
+AddBooking()
+UpdateBooking()
+DeleteBooking()
+ViewBooking()
Manage FAQ
FAQ
+ FAQId
+ FAQQuestion
+ FAQAnswer
+AddFAQ()
+UpdateFAQ()
+DeleteFAQ()
+ViewFAQ()

4.3.2 ER Diagram:-

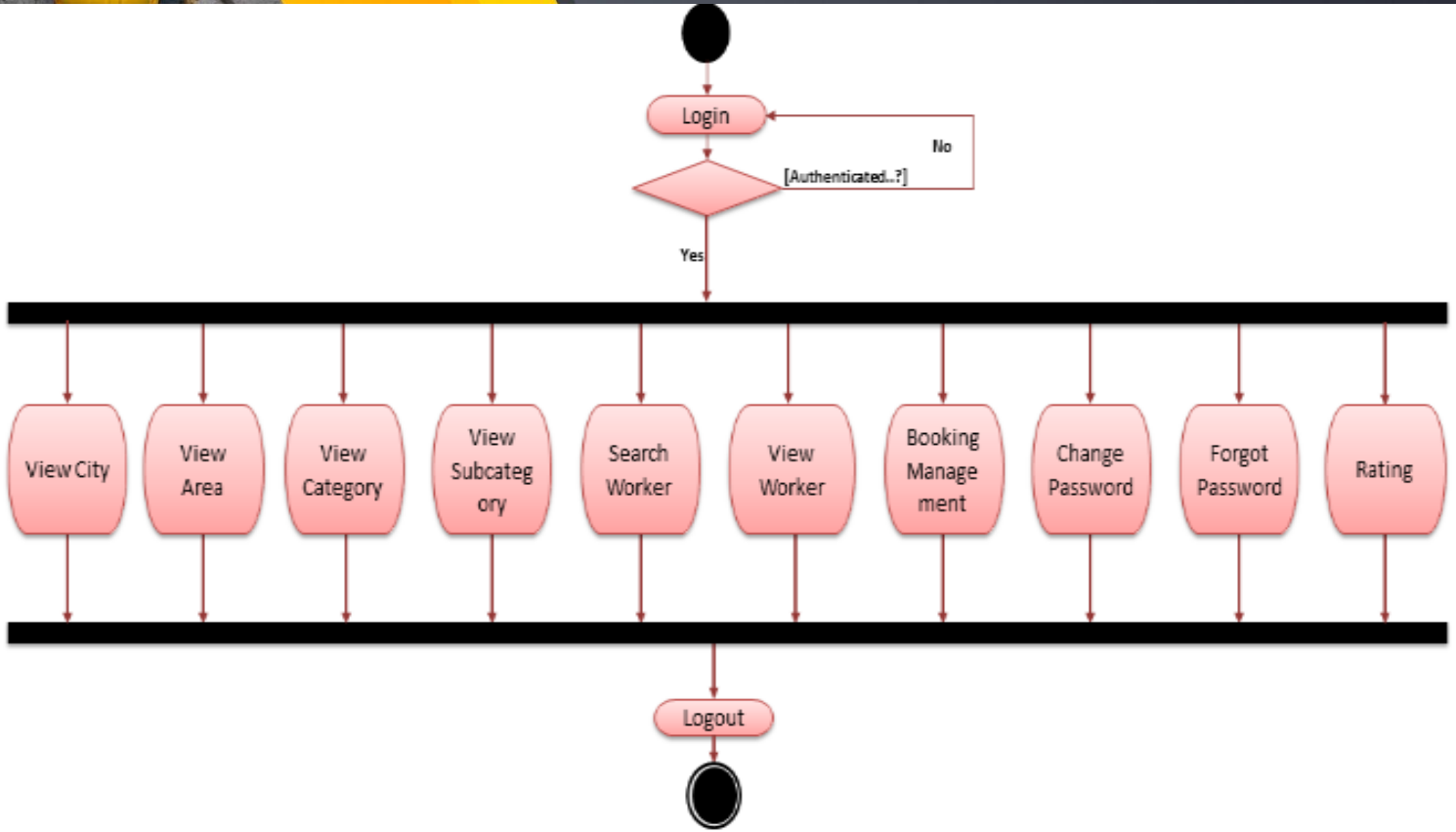

Send / Receive
Chat_id
Chat
Send / Receive
Client_id
Client_name
Client_address
User
Client_mobileno
Chat_date
Chat_description
Client_email
Gives
Admin_email
Admin_Name
Admin_id
Admin_Password
Admin
Adds
Area
Payment_amount
Payment_mode
Payment_date
Payment_id
Payment
Gives
Adds
Has
Has
Adds
Category_id
Category_name
Category
Has
Booking ID
Date
Details
Status
Booking
Adds
Id
Name
Service
Employee_id
Employee_name
Employee_address
Workerr
has
Employee_mobile
Employee_gender
Employee_email
Employee_password
Employee_pic

4.3.3 Activity Diagram

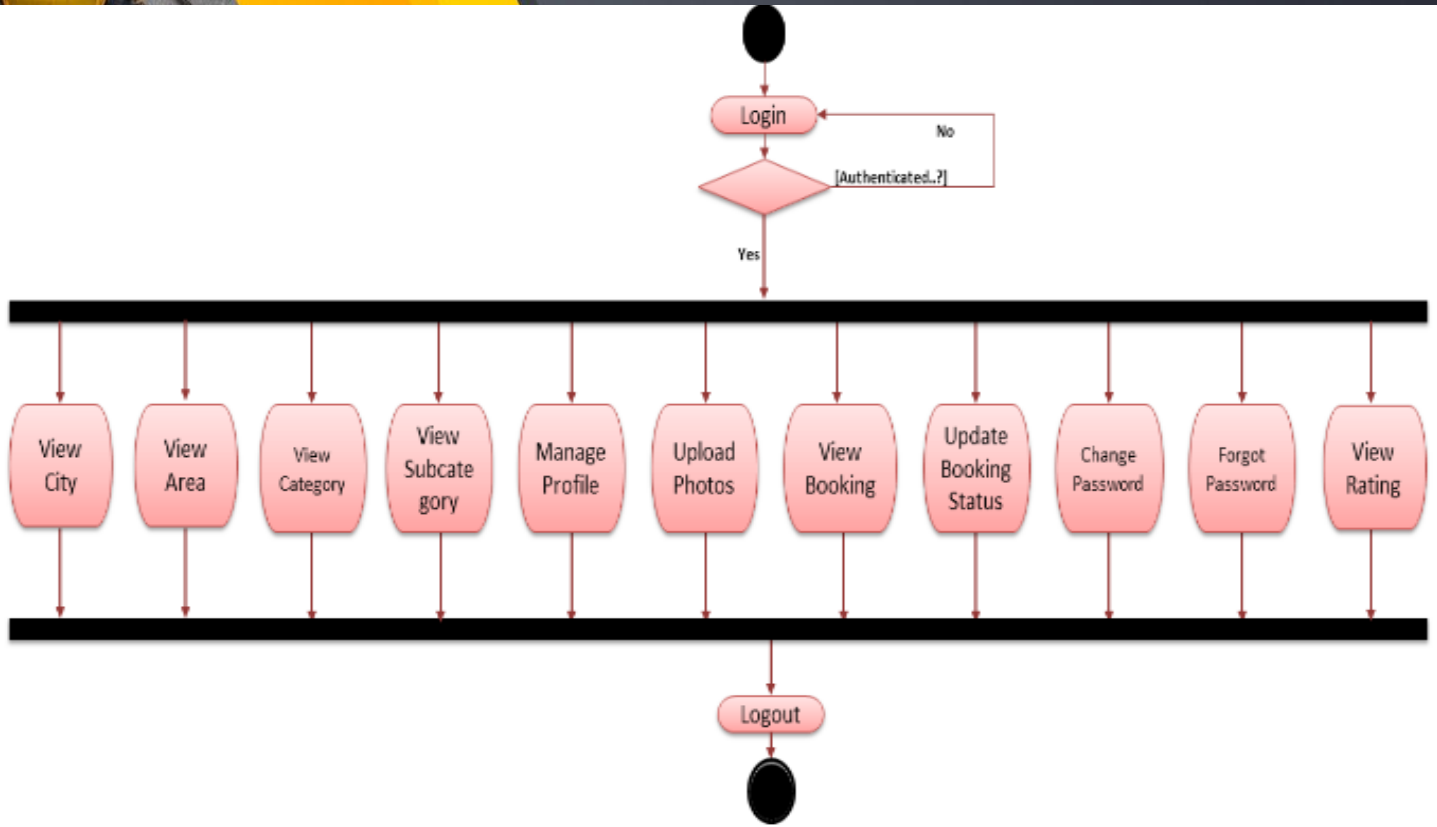
- Activity Diagram For Admin


Login
No
[Authenticated..?]
Yes
City Management
Area Management
Category Management
Subcategory Management
Search Worker
View Worker
View Booking
Change Password
Forgot Password
View Rating
Logout

Activity Diagram for User

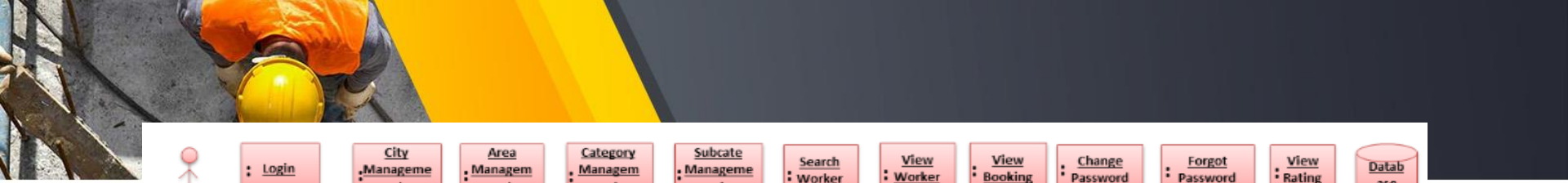


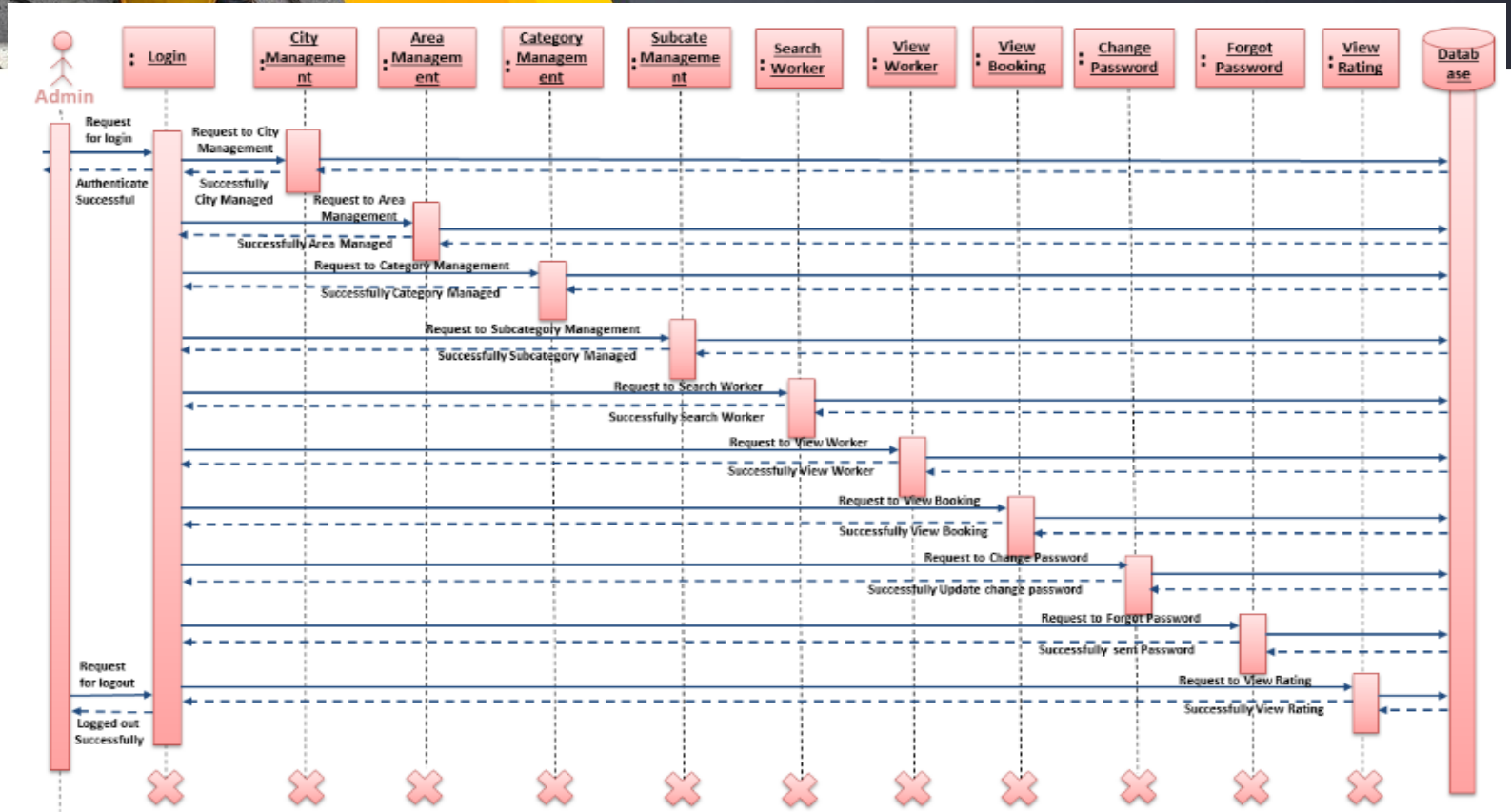
Activity Diagram for Worker



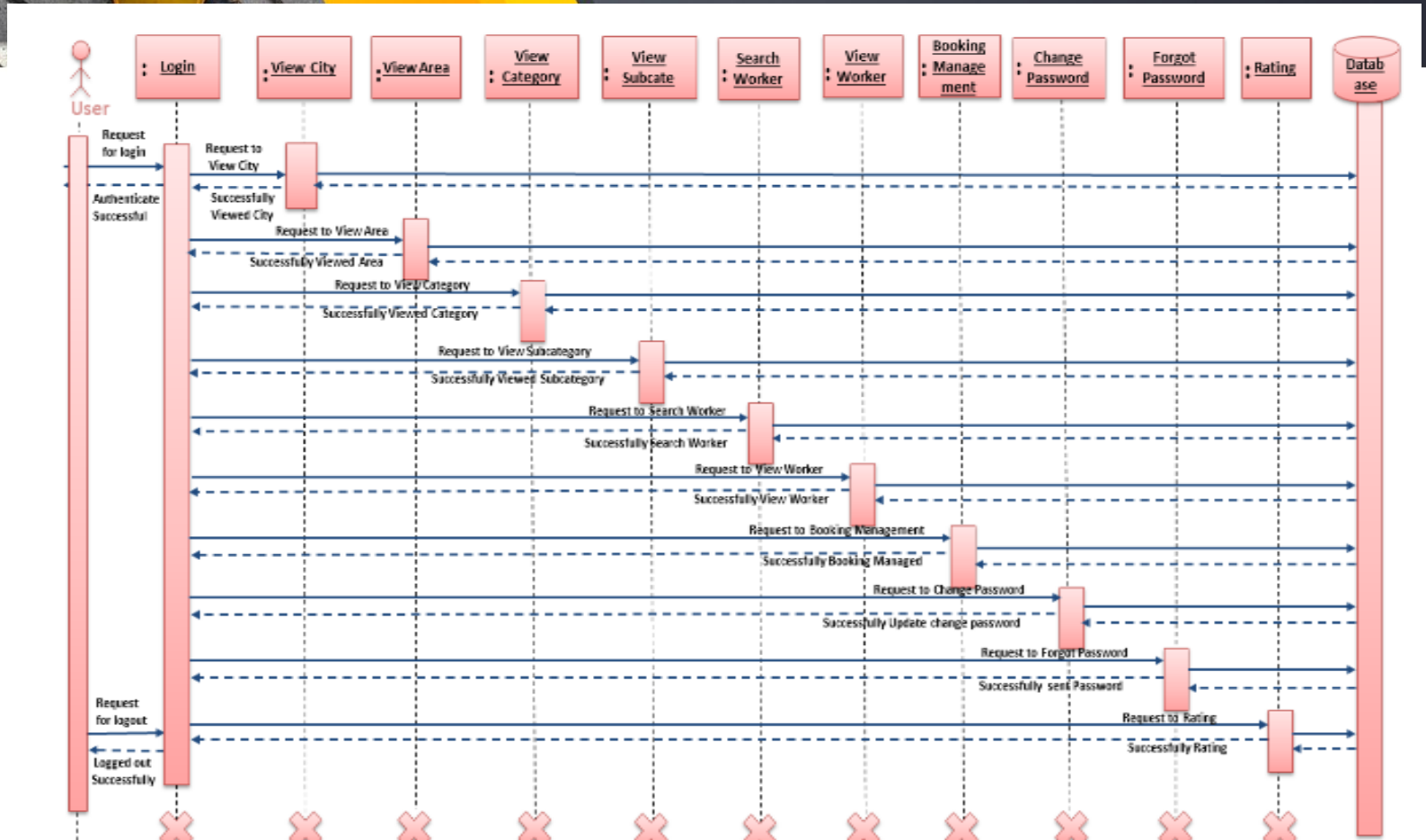
4.3.4 Sequence Diagram:-

- Sequence Diagram for Admin

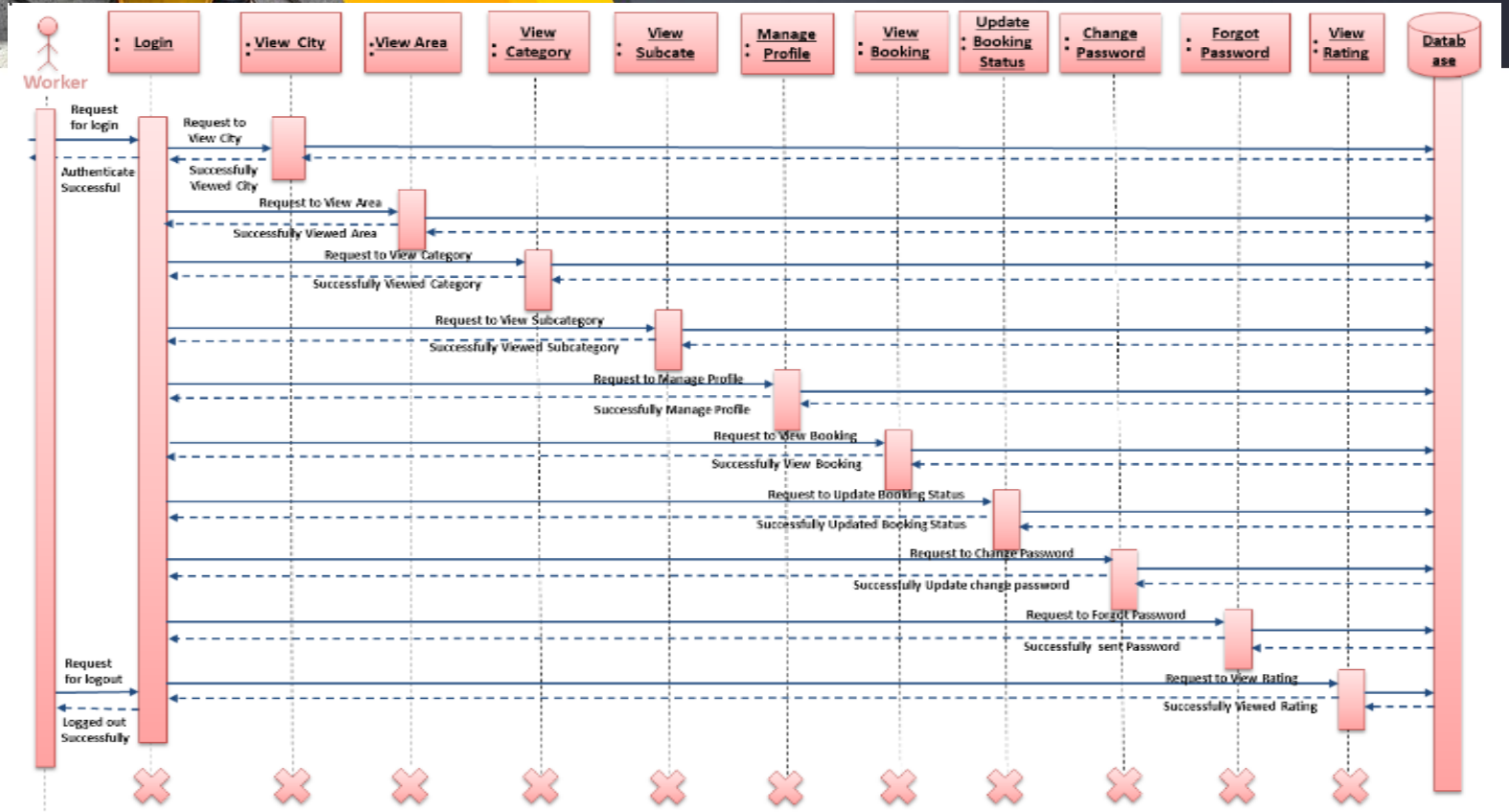



Admin
: Login
City :Management
Area :Management
Category :Management
Subcate :Management
: Search Worker
: View Worker
: View Booking
: Change Password
: Forgot Password
: View Rating
Database
Request for login
Request to City Management
Authenticate Successful
Successfully City Managed
Request to Area Management
Successfully Area Managed
Request to Category Management
Successfully Category Managed
Request to Subcategory Management
Successfully Subcategory Managed
Request to Search Worker
Successfully Search Worker
Request to View Worker
Successfully View Worker
Request to View Booking
Successfully View Booking
Request to Change Password
Successfully Update change password
Request to Forgot Password
Successfully sent Password
Request to View Rating
Successfully View Rating
Request for logout
Logged out Successfully

Sequence Diagram for User

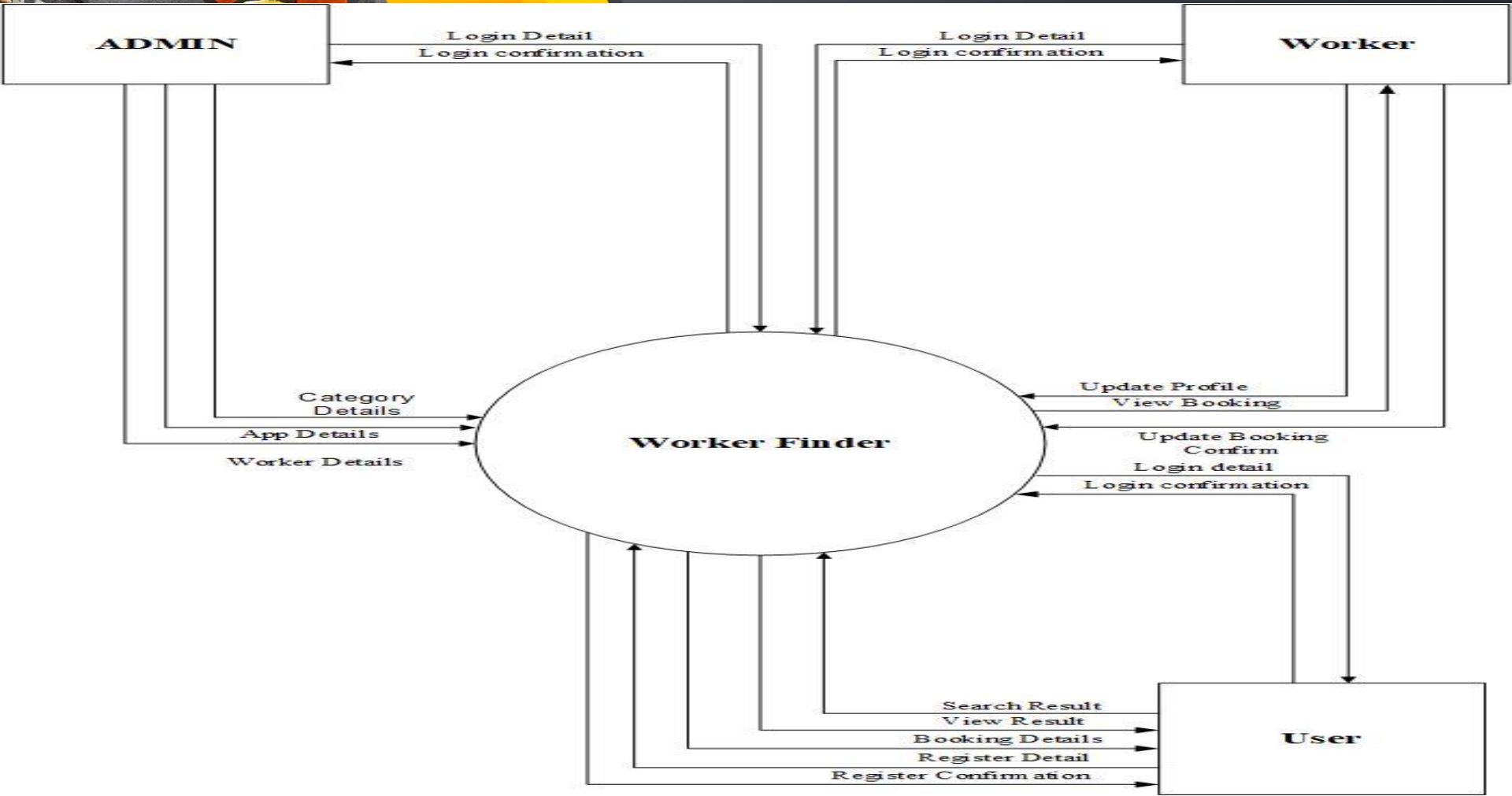


Sequence Diagram for Worker

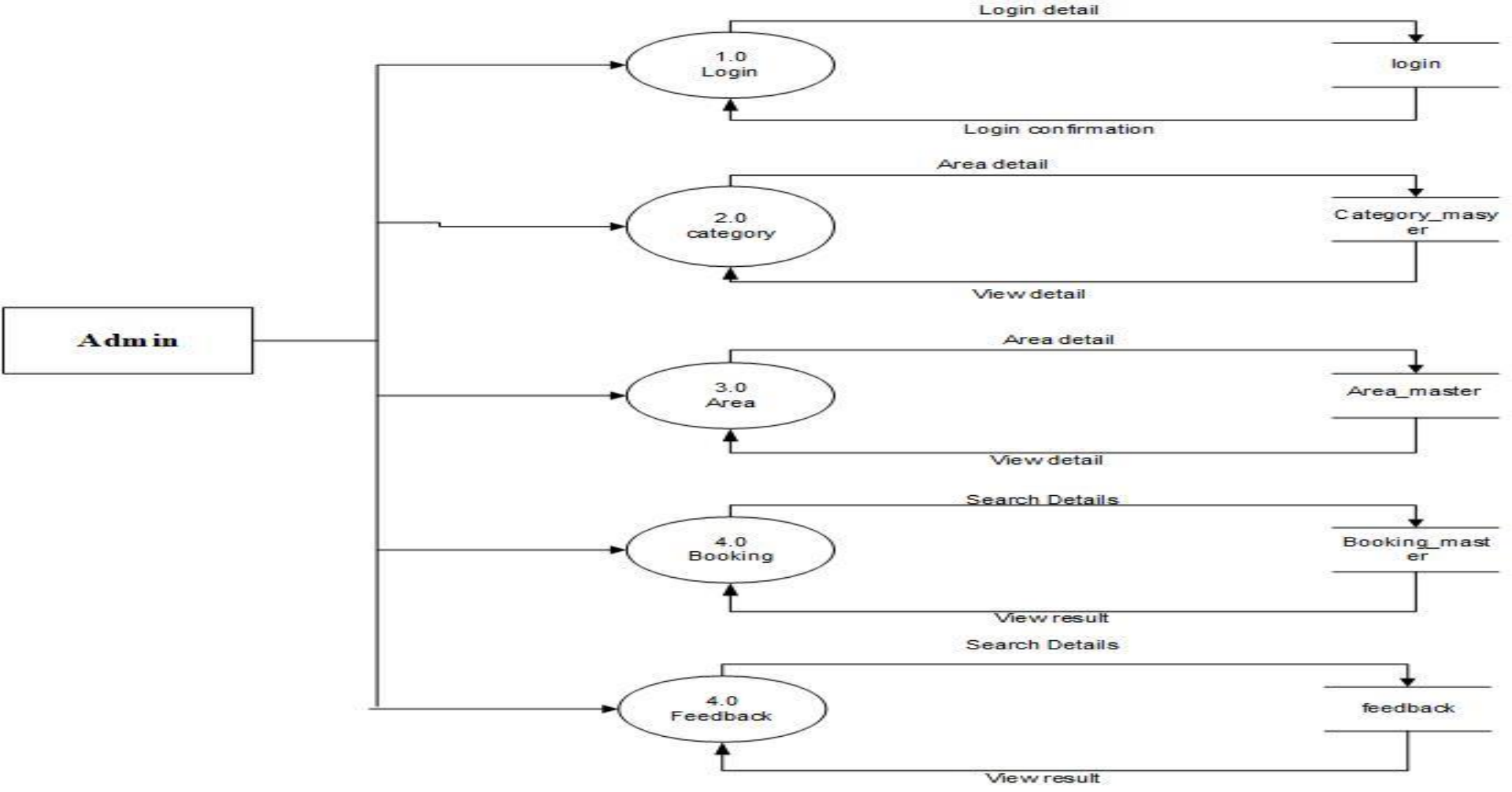


Data Flow

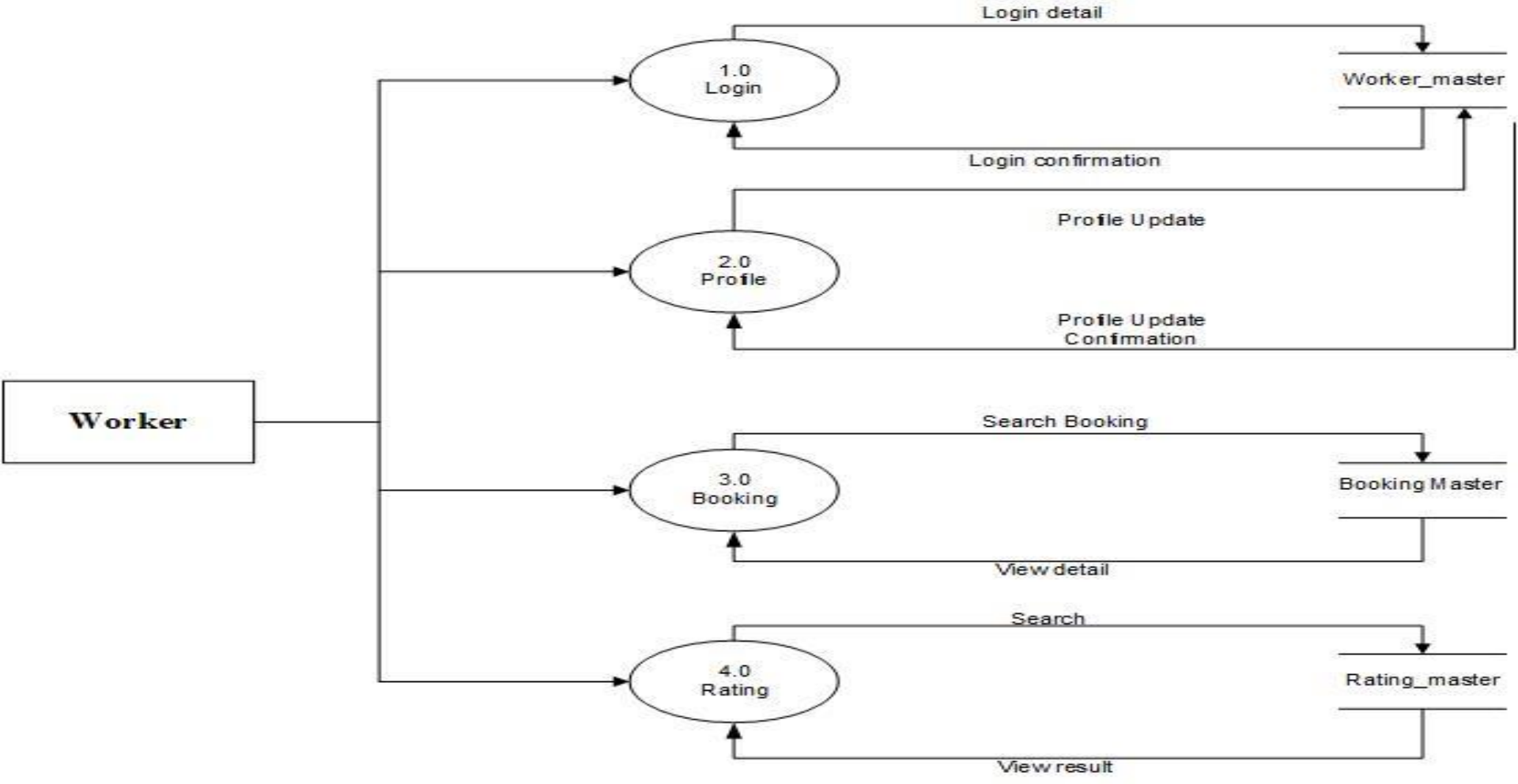
DFD Level 0


ADMIN
Login Detail
Login confirmation
Login Detail
Login confirmation
Worker
Category Details
App Details
Worker Details
Worker Finder
Update Profile
View Booking
Update Booking Confirm
Login detail
Login confirmation
Search Result
View Result
Booking Details
Register Detail
Register Confirmation
User

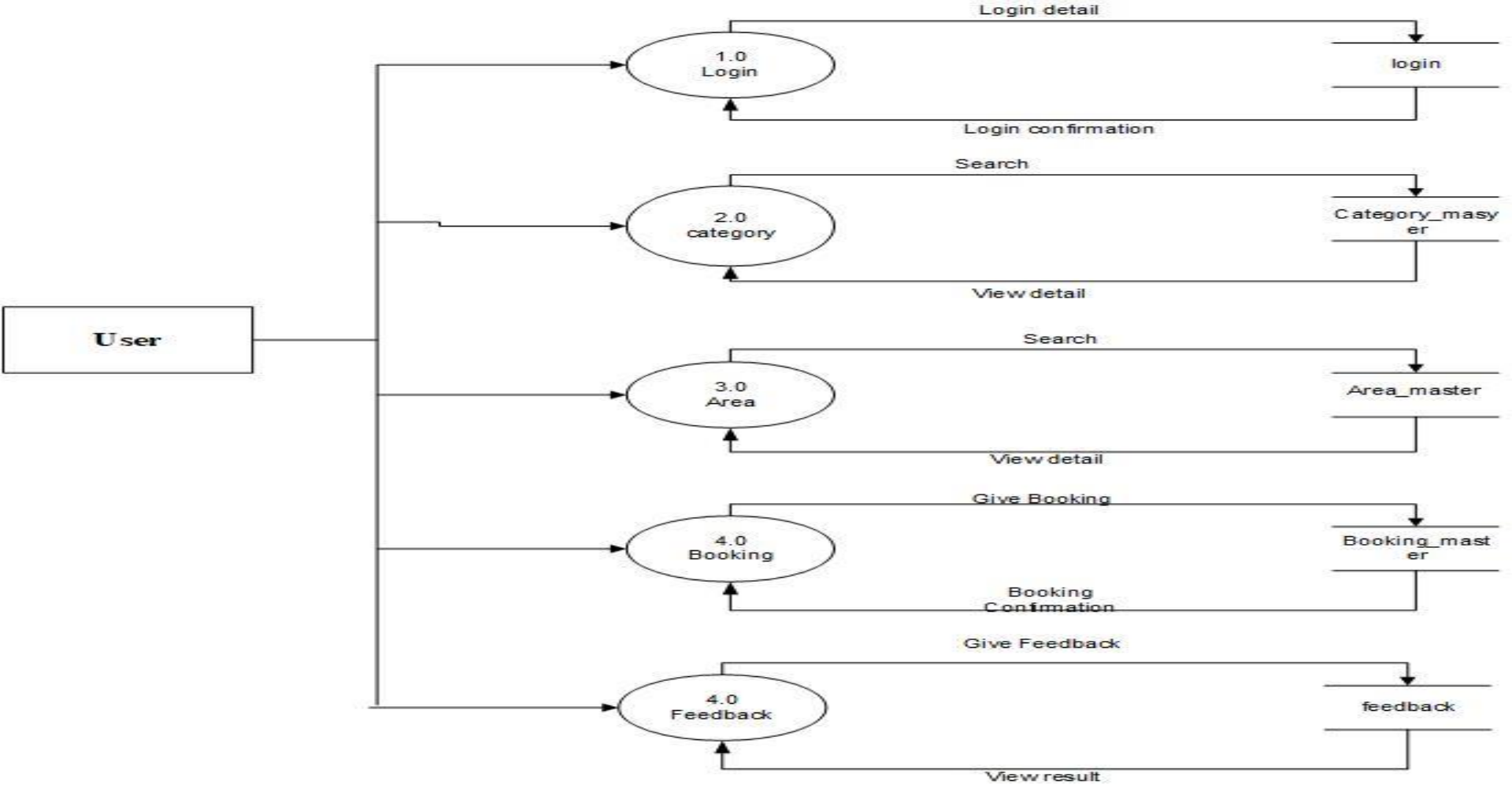
DFD Level 1


Admin
1.0
Login
Login detail
login
Login confirmation
2.0
category
Area detail
Category_masyer
View detail
3.0
Area
Area detail
Area_master
View detail
4.0
Booking
Search Details
Booking_master
View result
4.0
Feedback
Search Details
feedback
View result

DFD Level 2

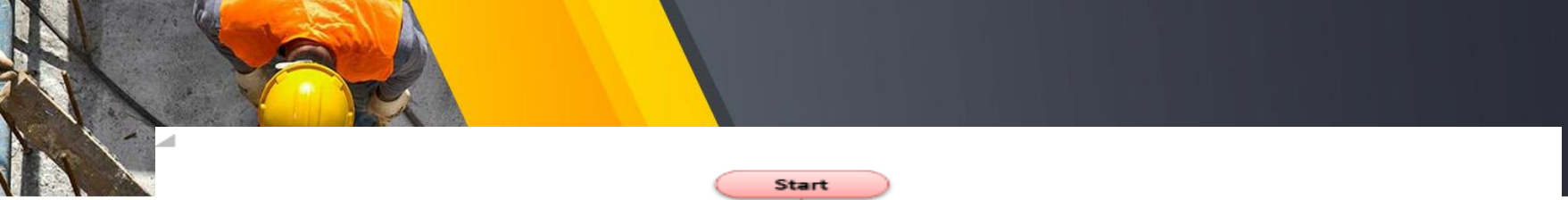

Login detail
1.0
Login
Worker_master
Login confirmation
Profile Update
2.0
Profile
Profile Update
Confirmation
Worker
Search Booking
3.0
Booking
Booking Master
View detail
Search
4.0
Rating
Rating_master
View result

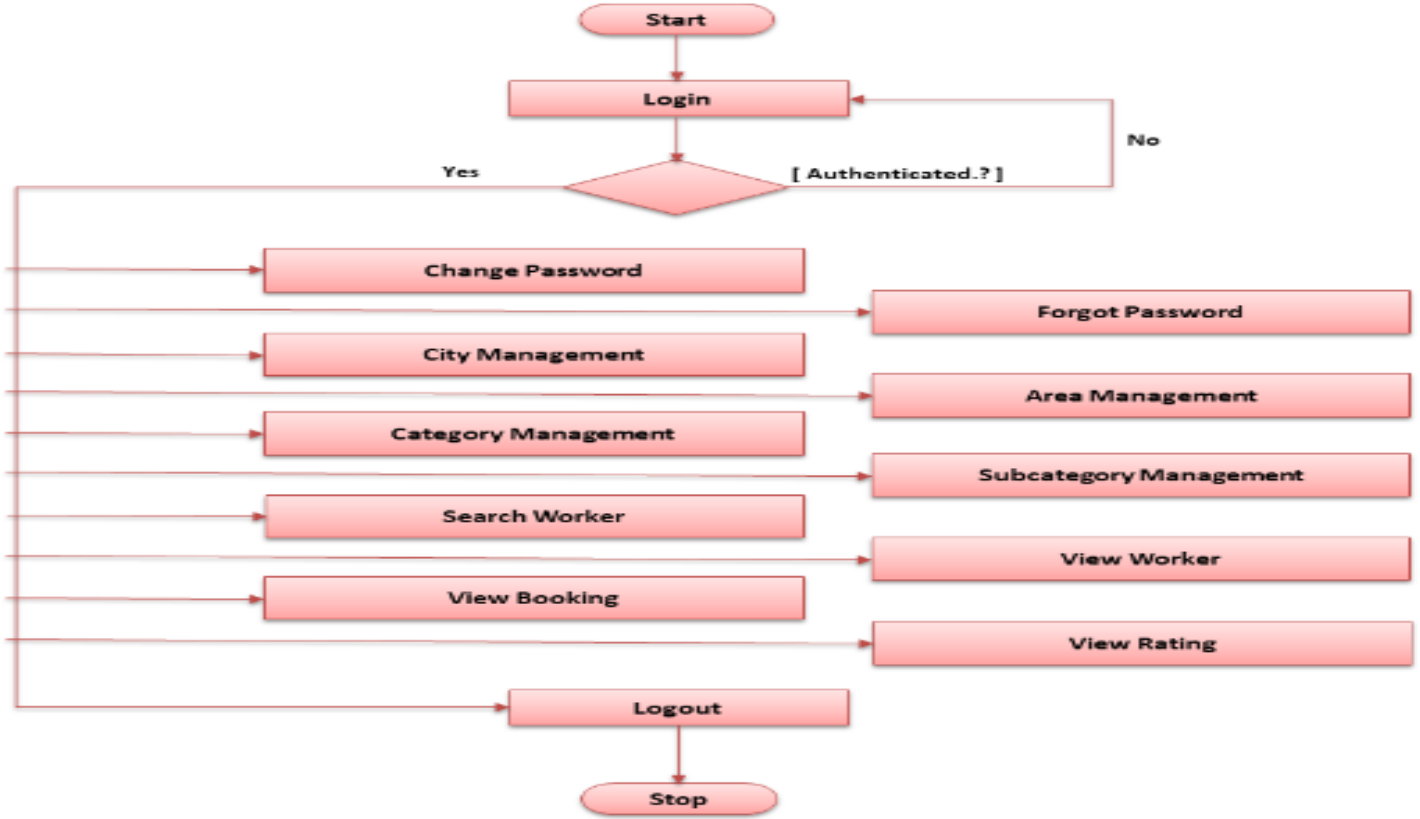
DFD Level 3


Login detail
1.0
Login
login
Login confirmation
Search
2.0
category
Category_masy
er
View detail
User
Search
3.0
Area
Area_master
View detail
Give Booking
4.0
Booking
Booking_mast
er
Booking
Confirmation
Give Feedback
4.0
Feedback
feedback
View result

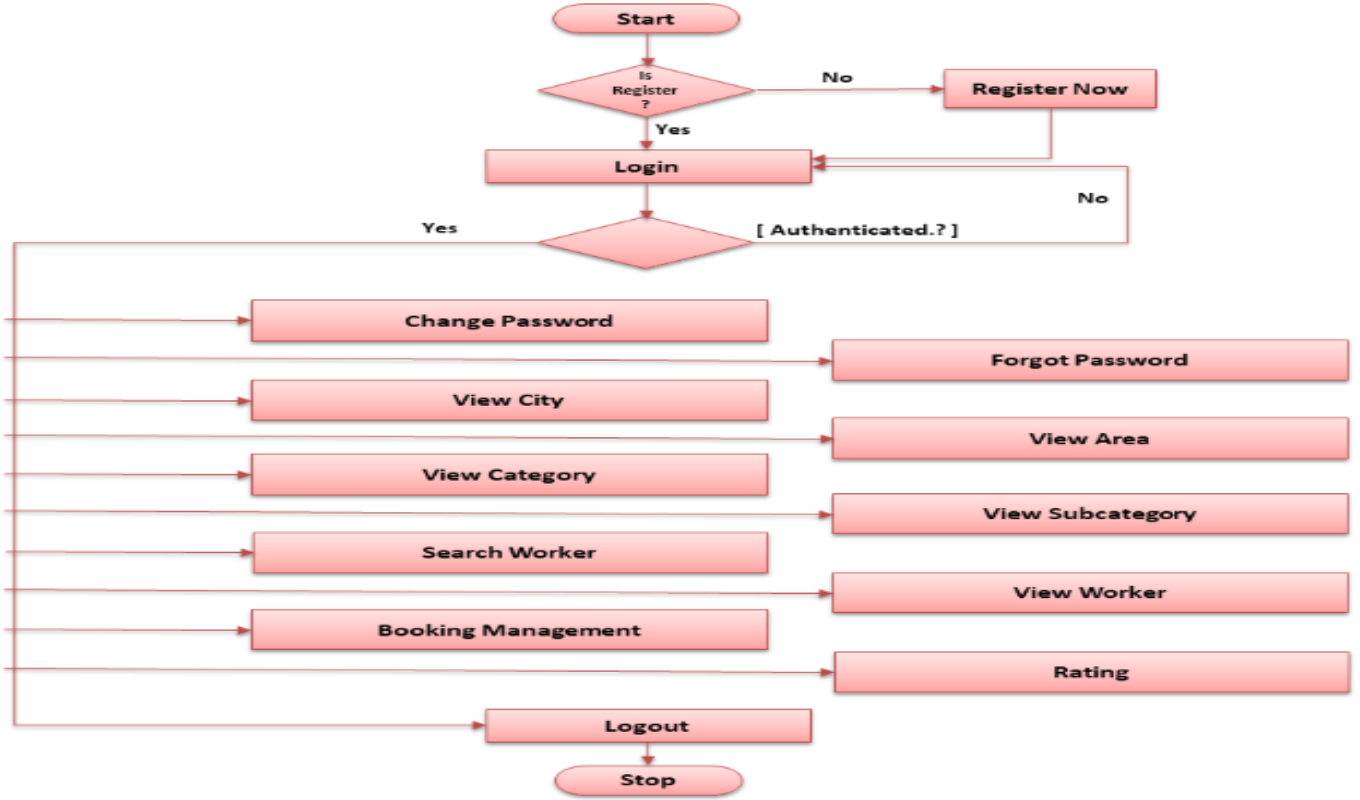
4.4.2 Flow Chart:-

- System Flow Chart for Admin

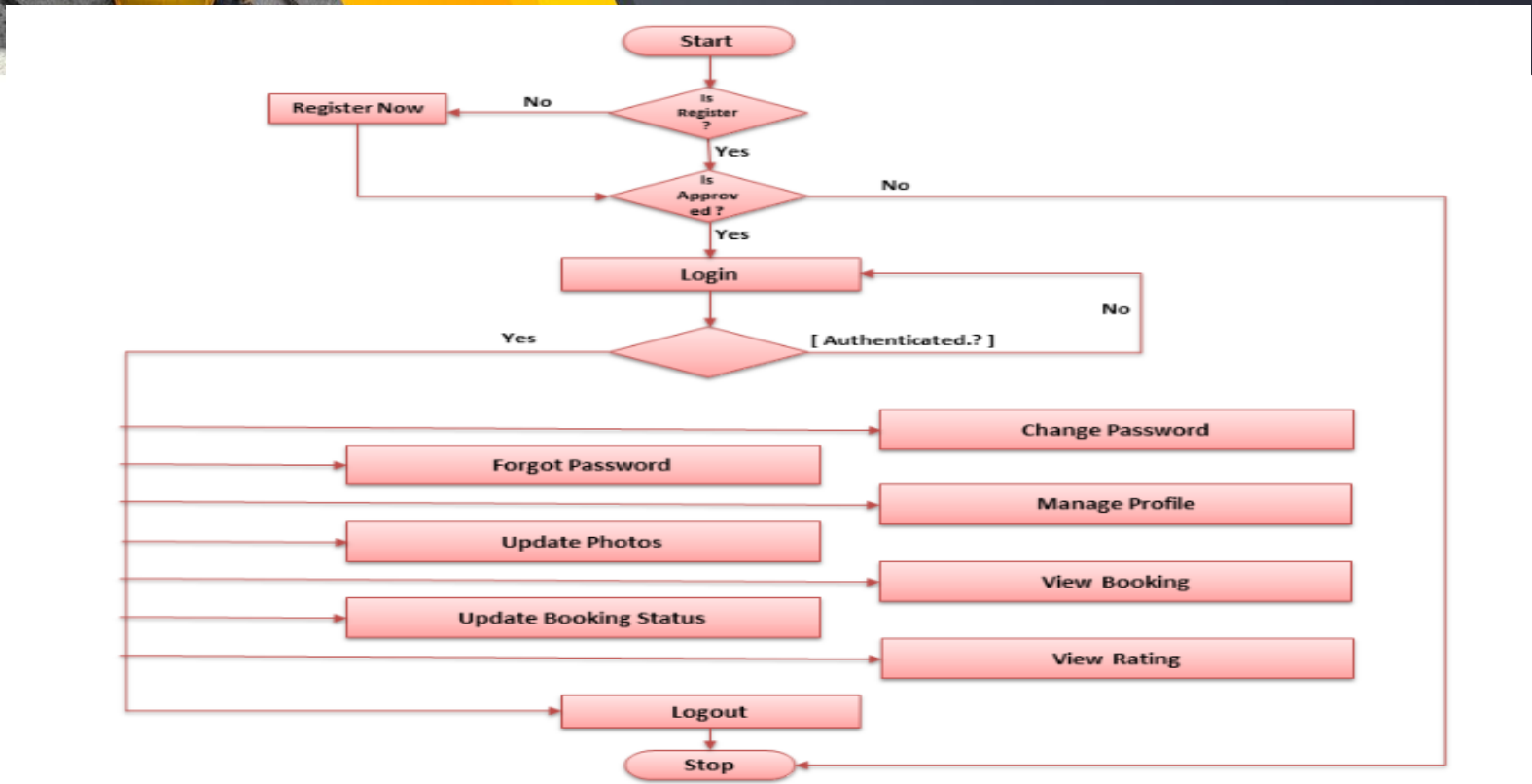



Start
Login
No
Yes
[Authenticated.?]
Change Password
Forgot Password
City Management
Area Management
Category Management
Subcategory Management
Search Worker
View Worker
View Booking
View Rating
Logout
Stop

System Flow Chart for User



System Flow Chart for Worker



CHAPTER 5 (DATA STRUCTURE)

5.1 Database Schema Design:-

Admin Description :- Represents Admin's Details.

Constraint	Field Name	Data Type	Description
Primary Key	admin_id	Int(11)	Admin ID
	Admin_name	Varchar(200)	Admin Name
	Admin_email	Varchar(200)	Admin Email
	admin_password	Varchar(200)	Admin Password
	admin_profile _pic	Varchar(200)	Admin Profile Pic
	is_active	Int(11)	Active or not

City Description :- Represents city Details.

Constraint	Field Name	Data Type	Description
Primary Key	city_id	Int(11)	City ID
	city_name	Varchar(200)	City Name
	is_active	Int(11)	Active or not

Area Description :- Represents Area's Details.

Constraint	Field Name	Data Type	Description
Primary Key	area_id	Int(11)	Area ID
	area_name	Varchar(200)	Area Name
Foreign key	city_id	Int(11)	City ID
	is_active	Int(11)	Active or not

User Master Description :- Represents User Details.

Constraint	Field Name	Data Type	Description
Primary Key	user_id	Int(11)	User ID
	user_name	Varchar(200)	User Name
	user_gender	varchar(10)	User gender
	user_mobile	bigint(20)	User Mobile No.
	user_email	Varchar(200)	User Email
	user_password	Varchar(200)	User Password
	address	Varchar(200)	User Address
Foreign key	city_id	Int(11)	City ID
Foreign key	area_id	Int(11)	Area ID
	user_photo	Varchar(200)	User Photo
	is_active	Int(11)	Active or not

Category Master

Description :- Represents Category's Details.

Constraint	Field Name	Data Type	Description
Primary Key	category_id	Int(11)	Category ID
	category_name	Varchar(200)	Category Name
	category_image	Varchar(200)	Category Image
	is_active	Int(11)	Active or not

Sub Category

Description :- Represents Subcategory Details.

Constraint	Field Name	Data Type	Description
Primary Key	sub_category_id	Int(11)	Sub Category ID
	sub_category_name	Varchar(200)	Sub Category Name
Foreign key	category_id	Int(11)	Category ID
	sub_category_image	Varchar(200)	Sub Category Image Path
	is_active	Int(11)	Active or not

Worker Master

Description :- Represents Worker's Details.

Constraint	Field Name	Data Type	Description
Primary Key	worker_id	Int(11)	Worker ID
	worker_name	Varchar(200)	Worker Name
	worker_gender	varchar(20)	Worker Gender
	worker_dob	date	Worker DOB
	email_id	varchar(200)	Worker Email ID
	mobile_no	varchar(11)	Worker Mobile No.
	password	varchar(11)	Worker Password
	worker_photo	varchar(200)	Worker Photo
	address	varchar(200)	Worker Address
Foreign key	city_id	int(11)	City ID
Foreign key	sub_category_id	int(11)	Sub Category ID

Foreign key	area_id	int(11)	Area ID
	aadhar_card	int(30)	Worker Aadhar Card No.
	pan_card	int(30)	Worker Pan Card No.
	worker_charge	int(11)	Worker Charge
	about_details	varchar(1000)	Worker Details
	experiance_detail	varchar(200)	Worker Experience Details
	business_name	varchar(200)	Worker Business Name
	is_block	int(11)	Worker Is Block Or Not
	is_available	int(11)	Worker Is Available Or Not
	is_online	int(11)	Worker Is Online Or Not
	is_verify	int(11)	Worker Is Verify Or Not
	is_active	Int(11)	Active or not

Booking_Master

Description :- Represents Booking Details.

Constraint	Field Name	Data Type	Description
Primary Key	booking_id	Int(11)	Booking ID
	booking_date	Timestamp	Booking Date
Foreign key	worker_id	Int(11)	Worker ID
	charge	Int(11)	Worker Charge
Foreign key	user_id	Int(11)	User ID
	address	varchar(200)	Address of User
Foreign key	area_id	Int(11)	Area ID
	is_complete	Int(11)	Service is Complete or not
	for_booking_time	varchar(100)	Time of Service
	for_booking_date	date	Date of Service
	is_active	Int(11)	Active or not

Worker Inquiry

Description :- Represents Worker inquiry Details.

Constraint	Field Name	Data Type	Description
Primary Key	worker_inquiry_id	Int(11)	Worker Inquiry ID
Foreign key	worker_id	Int(11)	Worker ID
Foreign key	user_id	Int(11)	User ID
	worker_inquiry_date	date	Worker Inquiry date
	is_active	Int(11)	Active or not

Worker Review

Description :- Represents Worker's Photo Details.

Constraint	Field Name	Data Type	Description
Primary Key	worker_review_id	Int(11)	Worker Review ID
Foreign key	worker_id	Int(11)	Worker ID
	rating	float	Worker Ratings
	review	varchar(200)	Worker Reviews
Foreign key	booking_id	Int(11)	Booking ID
	is_active	Int(11)	Active or not

Payment

Description :- Represents Payment Details

Constraint	Field Name	Data Type	Description
Primary Key	payment_id	Int(11)	Payment ID
	payment_date	date	Payment Date
Foreign key	booking_id	Int(11)	Booking ID
	amount	Int(11)	Amount of Payment
	method	varchar(200)	Method of Payment
	is_active	Int(11)	Active or not

FAQ

Description :- Represents FAQ Details.

Constraint	Field Name	Data Type	Description
Primary Key	faq_id	Int(11)	FAQ ID
	faq_question	Varchar(1000)	FAQ Question
	faq_answer	Varchar(1000)	FAQ Answer
	is_active	Int(11)	Active or not

CHAPTER 6 (Conclusion)

I have tried my best to make project perfect as per the users requirements. So users can find a best worker nearest his/her are.

Future Work:-

I will do implementation in our next phase of project or 6th semester duration.

Future Enhancements

Enhancements and improvement is continuously ongoing process for the app's lifetime, so to enhance the functionality and make it more usable for the user.

Here are the some future enhancements which will be done on the app:

- By Extending the use of the Free time, suggest the user for best encounter options according to the location of high priority contact and user
- Add multi language support for the application
- Add chat like functionality for the user to interact with the contact from the app itself.

CHAPTER 7(BIBLIOGRAPHY)

Coding

http://stackoverflow.com/

https://www.google.co.in/

Design

https://in.pinterest.com/

http://www.freepik.com/

http://www.flaticon.com/