EASY TRACK

APPLICATION DEVELOPMENT – I

SOFTWARE REQUIREMENTS SPECIFICATION

Submitted in partial fulfilment for the award of the degree

Of

Master of Technology

in

Information Technology

By

MAYANK KUMAR

(15MIN0018)

School of Information Technology and Engineering

September, 2017

Table of Contents

1. INTRODUCTION

Easy Track is a web application for planning and monitoring projects. It keeps track of project plans, project progress, manage budgets and more. It also gives project managers and business owners the control of managing multiple contractors and consultants. All team members have access to the project plan. Company leaders, however, ensure that team members only can access the information they need with role-based authorization system and user permissions.

1.1 PURPOSE

One of the most significant changes for the business world in the 21st century has been the move to a more remote workforce, as opposed employing the traditional in-house workforce. My project is based on such a requirement where we are trying to provide faster means of communication and information exchange. Today, companies face the challenge of tracking and managing outside resources that are not punching a time clock or even setting foot in their headquarters. Yet, companies need to know that these outside contractors and consultants are putting in the time they report, working productively and efficiently, and remaining true to the project goals and company values.

1.2 SCOPE

This is a significant change compared to traditional project management tools, where all the work was centralized in the project manager, who could not spend his time gathering and inserting all the information required continuously.
Companies often have individuals and teams working on projects from remote locations, and it can be difficult for them to remain in contact. Easy Track is a tool that helps company leaders, team members, and independent contractors and consultants collaborate and communicate quickly and effectively.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

- ✓ HTML – Hyper Text Mark-up Language.
- ✓ XHTML – extensible Hypertext Mark-up Language.
- ✓ HTTP – Hypertext Transfer Protocol.
- ✓ JSP – Java Server Pages.
- ✓ Project Plan – A project plan, according to the Project Management Body of Knowledge, is a formal, approved document used to guide both project execution and project control. The primary uses of the project plan are to document planning assumptions and decisions, facilitate communication among stakeholders, and document approved scope, cost, and schedule baselines. (Wikipedia)

1.4 REFERENCES

Wikipedia.
Project Management JumpStart, 3rd Edition.
Brilliant Project Management: What the best project managers know, do and say (3rd Edition) by Stephen Barker.
Fundamentals of Software Project Management by Rajiv Chopra.

1.5 OVERVIEW

Easy Track is a tool to keep everyone involved in a project, from the business owner to the independent consultants, on the same page. Teams are more productive when they have the ability to communicate easily.
A specialized tool for marketing teams that need to strategize, execute, and deliver results, Easy Track is an end-to-end content marketing platform. The platform aids marketing teams with multiple contractors and consultants to work collaboratively across departments, regions, and product lines so that every member of the team knows what needs to be done and when.
It is a set of online modules intended to help professionals manage programs, projects, requirements, testing, and product development. Organizations are free to choose a single module or mix and match to get just the right combination of Easy Track modules for their business needs.

2. GENERAL DESCRIPTION

Today each and every thing is based on the Computer. The computerization helps in speeding up the process therefore helps in reducing time consumption. Still there are many Organizations and project managers' use the manual method of tacking various projects.

They mostly use MS Excel, it is very difficult maintain and track the data in that. There are chances of errors. Indeed it is a time consuming process. There will not be any options to get all the details of the project progress. Other users like clients, Developers, Architects, Testers and other team members do not have interface to view the particular project status.

To overcome this problem, I have planned to develop an application called as Easy Track.
This project management software is flexible and configurable enough to meet the needs of a wide array of project teams and types, including those that involve the management of multiple contractors and consultants.

This tool strengthens that all the team take part in the project planning and states that everybody should have access to the project plan. Easy Track turns this possible on needing just a browser to access to the application.

Besides, it has a role based authorization system that allows you to configure the wished user’s permissions to fit their position in the organization.
Without any doubt to know the daily status of the execution of the projects in a company is a great advantage for the project manager. Easy Track, as a collaborative project management tool, allows collecting all the tracking information of the projects in real time and at small effort. All the workers can do it interacting with the application.

Companies are not isolated ones from the others. Many times they collaborate to deliver a project on time. Easy Track is ready to share and update plans among different companies.
It stores project information in one repository to share across the organization, with user access rights. It also Eliminates spreadsheets and automate executive project and portfolio reports

With Easy Track one can configure the resources with skills and outstanding conditions to perform tasks that they satisfy through time. Jointly with criteria the tool provides powerful resource allocation strategies which allocate automatically the free and qualified resources by just specifying the required criteria. This is an unbeatable feature which saves a lot of time.

2.1 PRODUCT PERSPECTIVE

- One private, secure, centralized website.
- Collaborate with multiple team members on multiple projects.
- Track hours, measure progress, and calculate project costs
- Integrated view of project and financial information
- Qualitative and quantitative key performance indicators
- Budget analysis with planned and actual costs over time
- Risk and issue management
- keeps the team working together, regardless of their location
- Delegate tasks, share files, and discuss topics
- Preview all projects in the Timeline view
- Project organization with task management including due dates, category labels, priorities, and more
- Automatically assign tasks, update databases, send email reminders, and more to work productively
- Share and communicate with multiple contractors and consultants as easily as you do with in-house employees

2.2 PRODUCT FUNCTIONS

- Action items, document sign-offs, and project deliverables
- Task due dates and reporting
- Get powerful analytics and then draw the content roadmap based on results
- Largest number of integrations in the industry
- Prioritize projects in portfolios
- Merges the power of the cloud with social communications for real-time updates, insights, and collaboration
- Dashboards and reporting
- Demand and document management
- Project invoicing
- Social collaboration with connections to Microsoft Outlook and other business productivity applications
- Project discussion boards included in project workspace templates
- Manage schedules with a variety of scheduling methods and tools to meet each project manager's needs
- Mobile connectivity

2.3 USER CHARACTERISTICS

There are many eventual users who will use my application. The general characteristics of the eventual users of the product that will affect the specific requirements are as below:

1) **Project Manager:** The Project Manager ensures that the project is delivered on time, to budget and to the required quality standard (within agreed specifications). He/she ensures the project is effectively resourced and manages relationships with a wide range of groups (including all project contributors). The Project Manager is also responsible for managing the work of consultants, allocating and utilising resources in an efficient manner and maintaining a co-operative, motivated and successful team

Responsibilities:

- ✓ Managing and leading the project team.
- ✓ Recruiting project staff and consultants.
- ✓ Managing co-ordination of the partners and working groups engaged in project work.
- ✓ Detailed project planning and control including:
- ✓ Developing and maintaining a detailed project plan.
- ✓ Managing project deliverables in line with the project plan.
- ✓ Recording and managing project issues and escalating where necessary.
- ✓ Resolving cross-functional issues at project level.
- ✓ Managing project scope and change control and escalating issues where necessary.
- ✓ Monitoring project progress and performance.
- ✓ Providing status reports to the project sponsor.
- ✓ Managing project training within the defined budget.

- ✓ Liaises with, and updates progress to, project board/senior management.
- ✓ Managing project evaluation and dissemination activities.
- ✓ Managing consultancy input within the defined budget.
- ✓ Final approval of the design specification.
- ✓ Working closely with users to ensure the project meets business needs.
- ✓ Definition and management of the User Acceptance Testing programme.
- ✓ Identifying user training needs and devising and managing user training programmes.

2) **Senior Consultant or Supplier side Project Manager:** The person responsible for managing supplier-side input to the project.

Responsibilities:

- ✓ Ensures that mandatory supplier requirements are met.
- ✓ Manages the production and approval of the supplier side of the budget.
- ✓ Makes effective use of supplier resources within the approved budget.
- ✓ Tracks performance of consultants and takes appropriate action.
- ✓ Proactively develops a collaborative relationship with the organisation to Project Steering Board level.
- ✓ Ensures that there are clear communication paths within the project team and the organisation and supplier.
- ✓ Acts as main point of contact between the supplier and the organisation.
- ✓ Produces and monitors financial reports including entry and maintenance of all actual time and expense against the master plan.
- ✓ Day to day management of supplier staff assigned to the project.
- ✓ Quality Assures the work of supplier staff assigned to the project.
- ✓ Within the organisation.

3) **Project Team Members:** The staff who actively works on the project, at some stage, during the lifetime of the project. Some may have a specific role – for example, the Team might include a Project Administrator.

Responsibilities:

- ✓ Provide functional expertise in an administrative process
- ✓ Work with users to ensure the project meets business needs
- ✓ Documentation and analysis of current and future processes/systems
- ✓ Identification and mapping of information needs
- ✓ Defining requirements for reporting and interfacing
- ✓ User training

4) Project Administrator or Co-ordinator: Responsible for maintenance of the project plan, maintenance and updating of a project website (if appropriate). Provides administrative support to the Project Manager. This role is most likely to be required in larger cross-functional projects.

Responsibilities:

- ✓ Sets up and manages support functions covering planning, tracking, reporting, quality management and internal communication.
- ✓ Produces consolidated reporting to the Project Board, including milestone summary, key issues, risks, benefits, summary of costs incurred.
- ✓ Establishes standards, tools and procedures for use on the project, including Issue, Risk, Change and Information Management.
- ✓ Manages the Project Library.
- ✓ Reviews project activities for compliance with procedures and standards.
- ✓ Manages the support and provision of project tools and equipment.
- ✓ Manages data security, software and license control.
- ✓ Assists with the production of user documentation.
- ✓ Assists with testing.

5) Systems Developer: To work with the Project Manager on defining and executing development requirements.

Responsibilities:

- ✓ Working with the Project Manager on definition of development requirements and priorities.
- ✓ Data Migration.
- ✓ Interfaces with other systems.
- ✓ Reporting configuration and deployment.
- ✓ Set up and maintenance of security rights and access permissions.
- ✓ Contributing to technical strategy, policy and procedure.
- ✓ Development and operation of technical testing programmes.
- ✓ Production of technical documentation to agreed quality standards.
- ✓ Reporting on progress/issues to management and users.

6) System Administrator: Management and support of the IT system environments.

Responsibilities:

- ✓ Management and support of the various environments.
- ✓ Network operating systems management and support.
- ✓ Database management and support.
- ✓ Back-up and disaster recovery measures.
- ✓ Contributing to technical strategy, policy and procedure.
- ✓ Development and operation of technical testing programmes.
- ✓ Production of technical documentation to agreed quality standards.

7) **Programme Manager:** This role is relevant if there are several related projects.

Responsibilities:

- ✓ Overall management and co-ordination of the programme of projects.
- ✓ Contributing to strategy, policy and procedure.
- ✓ Management of supplier/contractual relationships.
- ✓ Budgetary control of the programme of projects.
- ✓ Monitoring of, and responding to, issues at the programme level.

2.4 GENERAL CONSTRAINTS

Time – This refers to the actual time required to produce a deliverable. This in this case, would be the end result of the project. Naturally, the amount of time required to produce the deliverable will be directly related to the amount of requirements that are part of the end result (scope) along with the amount of resources allocated to the project.

Cost – This is the estimation of the amount of money that will be required to complete the project. Cost itself encompasses various things, such as: resources, labour rates for contractors, risk estimates, bills of materials, et cetera. All aspects of the project that have a monetary component are made part of the overall cost structure.

Scope – These are the functional elements that, when completed, make up the end deliverable for the project. The scope itself is generally identified up front so as to give the project the best chance of success.

The major take-away from the Triple Constraint, being that it is a triangle, is that one cannot adjust or alter one side of it without in effect, altering the other sides. So for example, if there is a request for a scope change mid-way through the execution of the project, the other two attributes (cost and time) will be affected in some manner. How much or how little is dictated by the nature and complexity of the scope change. As an added example, if the schedule appears to be tight and the project manager determines that the scoped requirements cannot be accomplished within the allotted time, both cost AND time are affected.

Along with recognizing how the triple constraint functions, it is imperative that the project manager convey that information to the project stakeholders. Making sure everyone who is involved with the project recognizes the importance of the constraint will make discussions regarding the scope, time and cost far easier. In many cases, the stakeholders are likely to be the main reasons for scope creep or budget adjustments in a project. Having them aware up front of what the ramifications might be for any requested or mandated changes will make dialog easier in follow-up meetings and will also make them scrutinize their change requests more thoroughly rather than assuming that any change will have no issue on the project release cycle. Note that conveyance of the triple constraint to the stakeholders is best performed at the outset, likely during the formation of the initial project plan.

As the project manager, making sure that he stays on top of all the key attributes of the triple constraint will make the likelihood of project success that much higher. So be cognizant of any fluctuations to the key attributes, whether they are unexpected or requested.

2.5 ASSUMPTIONS AND DEPENDENCIES

- ✓ Administrator can also be a project manager for a project.
- ✓ A Project manager has the access to view progress of all the modules.
- ✓ Client will not have the access to all the modules.
- ✓ The overall cost of day-to-day activity will not increase.
- ✓ Overall economic conditions will stay the same.
- ✓ Percentage of resources allocation remains in the same rate.

3 SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla, Chrome, Safari or Netscape Navigator by which user can access to the system.
The user interface shall be implemented using any tool or software package like Java Applet, MS Front Page, and Apache Tomcat Server etc.

3.1.2 Hardware Interfaces

Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable. The processor should be PENTIUM III & above. The minimum capacity of the RAM should be 256 MB. The hard disc capacity should be 10 GB and above.

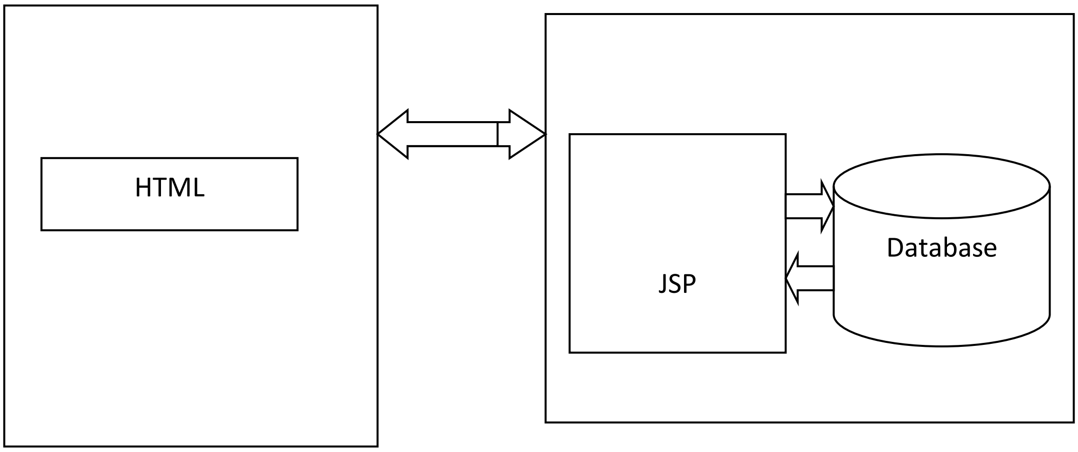
3.1.3 Software Interfaces

- ✓ The Easy Track system shall communicate to credit management system for handling financing options.
- ✓ The Easy Track system shall communicate with CRM system to provide support.
- ✓ Web Server - Apache Tomcat Server
- ✓ Browser - Internet Explorer/Mozilla Firefox/Chrome/Safari
- ✓ Server side scripting - JSP

- ✓ Database - MySQL
- ✓ Language - J2EE
- ✓ Client side scripting – HTML

SOFTWARE ARCHITECTURE: CLIENT SERVER

THE FOLLOWING DIAGRAM ILLUSTRATES THE LINKS BETWEEN VARIOUS COMPONENTS INVOLVED IN THIS SYSTEM:



In the following sections the salient features of each of their components are discussed:

- **JSP**
- **HTML**
- **JDBC**

Java Server Pages (JSP):

Java Server Pages (JSP) enables to separate the dynamic part of the pages from the static HTML. We simply write the regular html in the normal manner, using whatever Web-page-building tools you normally use. We then enclose the code for the c parts in special tags, most of which start with "<%" and end with "%>".
We normally give your file a .jsp extension, and typically install it in any ace you could place a normal Web page. Although what you write often looks more a regular html file than a servlet, behind the scenes, the JSP page just gets converted to a normal servlet, with the static html simply being printed to the output stream associated with the servlet's service method.
This is normally done the first time the page is requested, and developers can simply request the page themselves when first installing it if they want to be sure that the first real user doesn't get a momentary' delay when the JSP page is translated to a servlet and the servlet is compiled and loaded. Many Web servers let you define aliases that so that a URL that appears to reference an html file really points to a servlet or JSP page.

Composition:

Java Server Pages architecture could include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans and Servlets. As the default scripting language, Java Server Pages use the Java Programming language. This means that scripting on the server side can take advantage of the full set of capabilities that the Java programming language offers.

Processing:

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. It may have associated components in the form of .class, .jar, or .ser files- -or it may not. The use of components is not required.
The Java Server Pages file has a .jsp extension to identify it to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a servlet on the server side. The servlet that is generated, outputs real content in straight HTML for responding to the customer. Because it is standard HTML, the dynamically generated response looks no different to the customer browser than a static response.

Advantages of JSP:

- ✓ Separation of static from dynamic content
- ✓ Write Once Run Anywhere
- ✓ Recommended Web access layer for n-tier architecture
- ✓ Completely leverages the Servlet API
- ✓ Platform independent
- ✓ Reuse of components and tag libraries
- ✓ Encapsulation of functionality

- ✓ They have a better performance and scalability than ordinary CGI scripts, because they are persistent in memory and multi-threaded.
- ✓ They have built in support for HTTP sessions, which makes application Programming possible.
- ✓ They have full access to Java Technology-Network awareness, threads and Database connectivity-without the limitations of client side application applets.
- ✓ They are automatically recompiled when necessary.
- ✓ They exist in the ordinary Web server document space, no special is required to address them.

Java Script:

Java script is a fairly simple language, which is only suitable for fairly simple tasks. The language is best suited to tasks, which run for a short time, and is most commonly used to manipulate the pieces of the document object model. The idea behind finding JavaScript is to find a language which could be used to provide client-side in-browser applications but which was not as complicated as Java.

JavaScript is Netscape's cross-platform, object-oriented scripting language. Core JavaScript contains a core set of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. It is mainly used here for validation purpose.

Benefits of JavaScript:

JavaScript has a number of benefits to anyone who wants to make their Web site dynamic.

- ✓ It is widely supported in Web browsers.
- ✓ It gives easy access to the document objects and can manipulate most of them.
- ✓ JavaScript can give interesting animations without the long download times associated with many multimedia data types.
- ✓ Web surfers don’t need a special plug-in to use scripts.
- ✓ JavaScript is relatively secure- JavaScript can neither read from our hard drive nor write to it, and we can’t get a virus infection directly from JavaScript.

HYPERTEXT MARKUP LANGUAGE (HTML)

HTML means Hypertext Mark-up Language. HTML is a method of describing the format of documents which allows them to be viewed on computer screens. HTML documents are displayed by web browsers, programs which can navigate across networks and display a wide variety of types of information. HTML pages can be developed to be simple text or to be complex multimedia extravaganzas containing sound, moving images, virtual reality, and Java applets.

The global publishing format of the Internet is HTML. It allows authors to use not only text but also format that text with headings, lists, and tables, and to include still images, video, and sound within text. Readers can access pages of information from anywhere in the world at
The click of a mouse-button. Information can be downloaded to the reader's own PC or workstation. HTML pages can also be used for entering data and as the front-end for commercial transactions

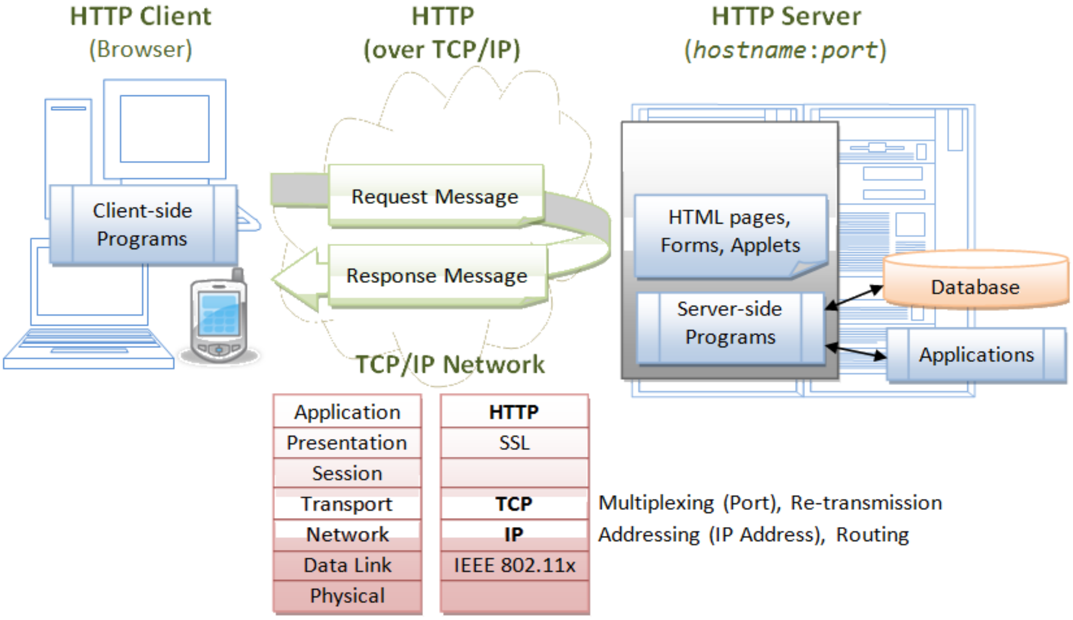
Features of HTML:

- ✓ It is not a programming language.
- ✓ It is not a data description language.
- ✓ It is simple to understand and implement.
- ✓ HTML constructs a very easy to comprehend, and can be used effectively by anybody.
- ✓ The methodology used by HTML to mark-up information is independent of its representation on a particular hardware or software architecture.
- ✓ HTML syntax is a worldwide standard.

3.1.4 Communications Interfaces

The notion of an **interface** in the work environment is used for a point of interaction between a number of systems or work groups.

The Easy Track system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.



3.2 FUNCTIONAL REQUIREMENTS

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

Functional Requirement Specifications:

The following are the key fields, which should be part of the functional requirements specifications document:

- Purpose of the Document
- Scope
- Business Processes
- Functional Requirements
- Data and Integration
- Data Migration & Conversion

EXAMPLE:

Once the project agreement is done, the project ID will be assigned to the project. This ID is unique for all the projects. Through this we can refer the projects. The useful information regarding project is collected from the user and it is maintained.

The reports will be generated on the basis of year, month and day. We can also see the project information on the basis of project ID and Company Name.

3.2.1 The system shall allow the administrator to add users.
3.2.2 The system shall allow the administrator to modify user information
3.2.3 The system shall allow the administrator to remove users.
3.2.4 The system shall allow the Project Manager to add users.
3.2.5 The system shall allow the Project Manager to allocate the project.
3.2.6 The system shall allow the Project Manager to de-allocate the project.
3.2.7 The system shall allow Project Manager to view other users' information.
3.2.8 The system shall allow the user to change the password.
3.2.9 The system shall allow users to view location information.

3.3 USE CASES

Use case diagrams are usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Note, that UML 2.0 to 2.4 specifications also described use case diagram as a specialization of a class diagram, and class diagram is a structure diagram.

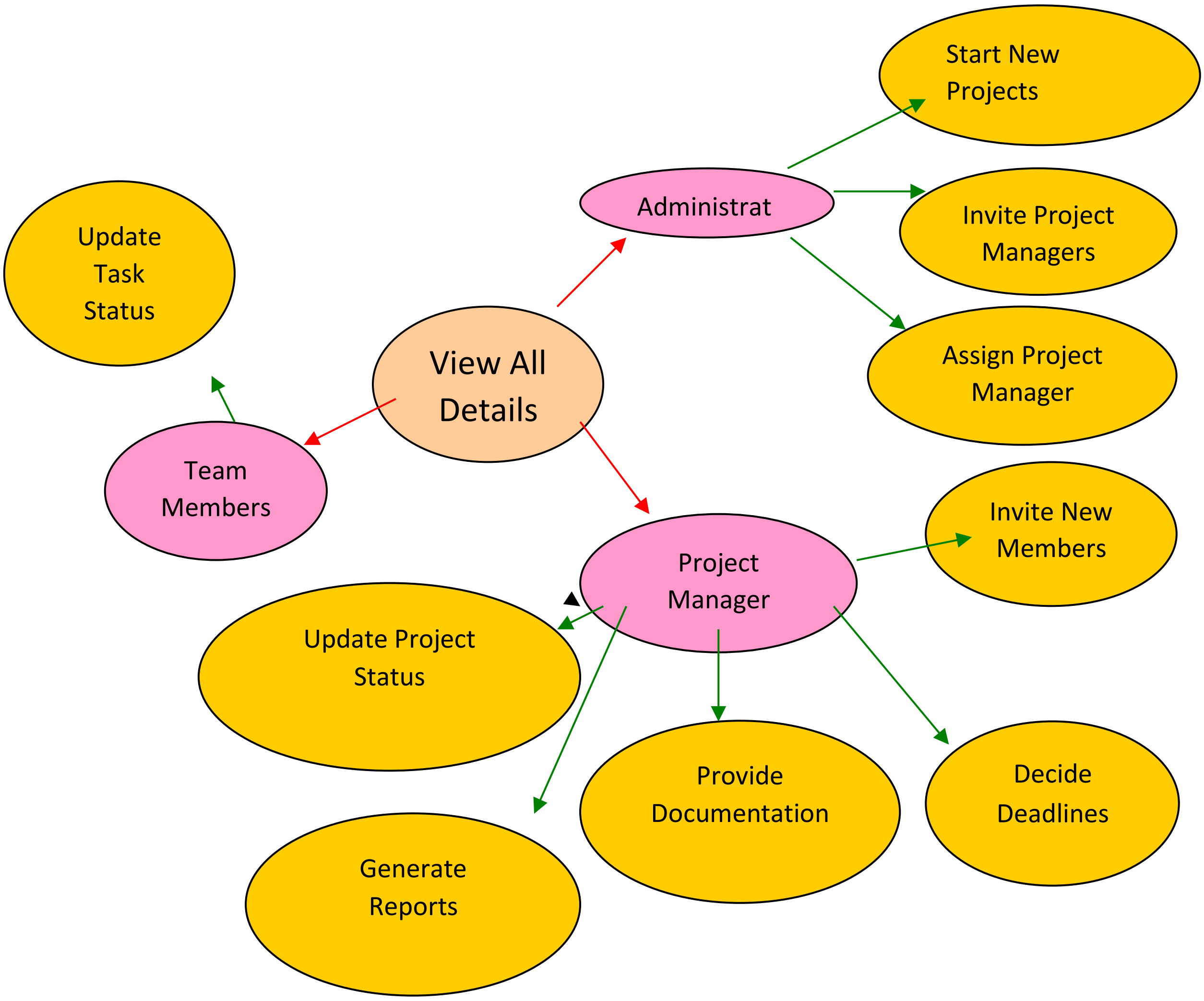
Use case diagrams are in fact twofold - they are both behaviour diagrams, because they describe behaviour of the system, and they are also structure diagrams - as a special case of class diagrams where classifiers are restricted to be either actors or use cases related to each other with associations.

[UML 2.5 FTF - Beta 1] moved use cases out of behaviour modelling to UML supplementary concepts. So, it is an unfortunate quandary what kind of UML diagrams use case diagrams are.

A use case (or set of use cases) has these characteristics:

- Organizes functional requirements.
- Models the goals of system/actor (user) interactions.
- Records paths (called scenarios) from trigger events to goals.
- Describes one main flow of events (also called a basic course of action), and possibly other ones, called exceptional flows of events (also called alternate courses of action).
- Is multi-level, so that one use case can use the functionality of another one.

EXAMPLE OF USE CASE DIAGRAM



1) **Administrator:** - is responsible for registering with the tool and creating a profile with the company name.

    a. Start New Project: - Admin logs into the OPM and create a profile with his company name and initiates a project.

    b. Invite Project Manager: - After starting a project the Admin Invites a Project Manager.

    c. Assign Project Manager: - When the invited Project Manager agrees to the Admin he/she is being allotted the project initiated.

2) **Project Manager:** - is responsible for dealing with all the proceedings of the project

    a. Invite New Team Member: - After the Project is being allotted a Manager, the PM invites various team members to work on the project as well and sends them invites.

    b. Provide Documentation: - The PM gives documentation for the various tasks.

    c. Generate Report: - The PM can generate reports for the various task and project.

    d. Decide Deadline: - The PM can decide the time needed for the task to be completed.

    e. Update Project Status: - The PM can update the completion status of the project on time-to-time basis.

3) **Team Members: -**

    a. Update Task Status: - the members working on the particular task can update the task completion status.

3.4 CLASSES / OBJECTS

An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time. The use of object diagrams is fairly limited, namely to show examples of data structure.

3.5 NON- FUNCTIONAL REQUIREMENTS

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements place restrictions on the product being developed, the development process, and specify external constraints that the product must meet.

3.5.1 Performance

The application should be based on web and has to be run from a web server.
The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run.
The performance shall depend upon hardware components of the client/customer.

3.5.2 Reliability

Software reliability is the probability that software will work properly in a specified environment and for a given amount of time.
The application should be reliable at any circumstances.

3.5.3 Availability

Availability is the ratio of time a system or component is functional to the total time it is required or expected to function. It can also be expressed in terms of average downtime per week, month or year or as total downtime for a given week, month or year. Sometimes availability is expressed in qualitative terms, indicating the extent to which a system can continue to work when a significant component or set of components goes down. The Easy Track application availability is constrained by the availability of the web application server, database server, and other supporting software servers.
My application should comply with the above availability criteria.

3.5.4 Security

- ✓ The system should use secure sockets in all transactions that include any confidential customer information.
- ✓ The system should automatically log out all customers after a period of inactivity.
- ✓ The system should confirm all transactions with the customer’s web browser.
- ✓ The system should not leave any cookies on the customer’s computer containing the user’s password.
- ✓ The system should not leave any cookies on the customer’s computer containing any of the user’s confidential information
- ✓ The system’s back-end servers should only be accessible to authenticated administrators.
- ✓ The system’s back-end databases shall be encrypted.

3.5.5 Maintainability

Maintainability includes concepts of modularity, understand ability, changeability, testability, reusability, and transferability from one development team to another. These do not take the form of critical issues at the code level. Rather, poor maintainability is typically the result of thousands of minor violations with best practices in documentation, complexity avoidance strategy, and basic programming practices that make the difference between clean and easy-to-read code vs. unorganized and difficult-to-read code. My application should be maintainable.

3.5.6 Portability

The system shall run on Windows Server 2003.
The system shall be compatible with IIS6.0 and above or Apache 2.0 and above.
The system shall conform to HTML standards and therefore support different web browsers including Internet Explorer, Firefox, Opera and Safari.

3.6 INVERSE REQUIREMENTS

Inverse requirements can be functional and non-functional.
When a customer specifies that something must not be done. For example, User ID should only contain digits. The application should cover Inverse requirements as well.

3.7 DESIGN CONSTRAINTS

The Easy Track application hardware limitations are defined by the limitations of its support web server and database server.
The Easy Track application software limitations are defined by the limitations of its support web server and database server.

3.8 LOGICAL DATABASE REQUIREMENTS

JAVA DATABASE CONNECTIVITY (JDBC)

JDBC AND ODBC IN JAVA:

Most popular and widely accepted database connectivity called Open Database Connectivity (ODBC) is used to access the relational databases. It offers the ability to connect to almost all the databases on almost all platforms. Java applications can also use this ODBC to communicate with a database. There are several reasons which needs JDBC

- ✓ ODBC API was completely written in C language and it makes an extensive use of pointers. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness and automatic portability of applications.
- ✓ ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries.
- ✓ ODBC drivers must be installed on client's machine.

ARCHITECTURE OF JDBC-ODBC:

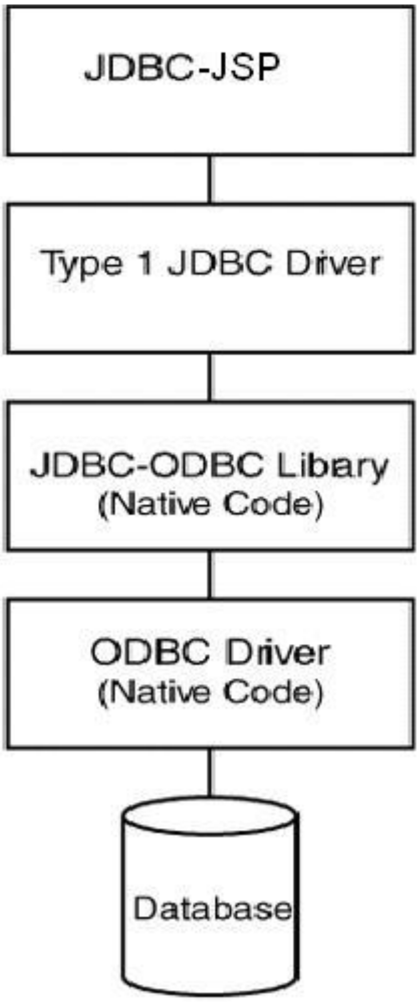
- ✓ Application Layer: Java program wants to get a connection to a database. It needs the information from the database to display on the screen or to modify the existing data or to insert the data into the table.

- ✓ Driver Manager: The layer is the backbone of the JDBC architecture. When it receives a connection-request form.

- ✓ The JDBC Application Layer: It tries to find the appropriate driver by iterating through all the available drivers, which are currently registered with Device Manager. After finding out the right driver it connects the application to appropriate database.

- ✓ JDBC Driver layers: This layer accepts the SQL calls from the application and converts them into native calls to the database and vice-versa. A JDBC Driver is responsible for ensuring that an application has consistent and uniform m access to any database

JDBC ARCHITECTURE CONTAINS FOLLOWING LAYERS:



When a request received by the application, the JDBC driver passes the request to the ODBC driver, the ODBC driver communicates with the database and sends the request and gets the results. The results will be passed to the JDBC driver and in turn to the application. So, the JDBC driver has no knowledge about the actual database, it knows how to pass the application request of the ODBC and get the results from the ODBC.

The JDBC and ODBC interact with each other. The reason is both the JDBC API and ODBC are built on an interface called “Call Level Interface” (CLI). Because of this reason the JDBC driver translates the request to an ODBC call. The ODBC then converts the request again and presents it to the database. The results of the request are then fed back through the same channel in reverse.

3.9 Other Requirements

There are no additional requirements. All the requirements mentioned above will be used to create the application.

4. ANALYSIS MODELS

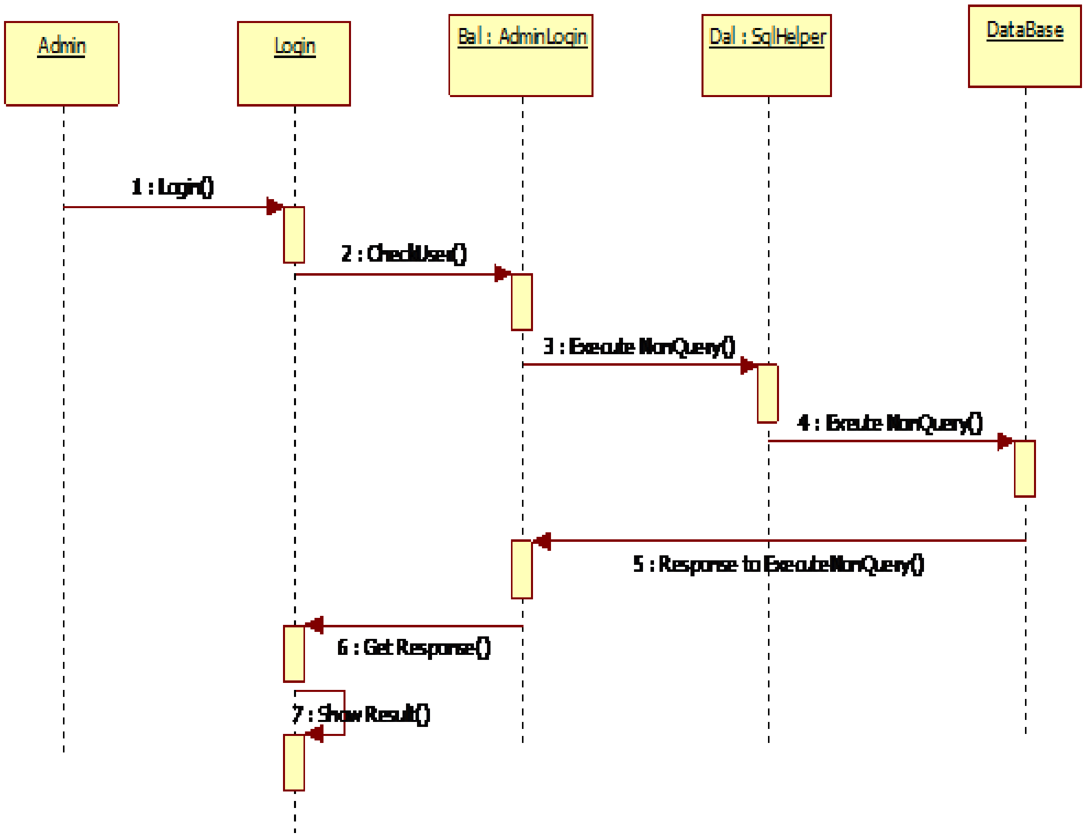
Requirement Analysis models act as the bridge between functional requirements and the detailed design of the software system. For example, Use cases lead to user interface design, data dictionary and entity relationship diagrams are used for designing database schema and class diagrams.

4.1 Sequence Diagrams

A **Sequence diagram** is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

4.1.1 Sequence Diagram



4.1.1 Sequence diagram

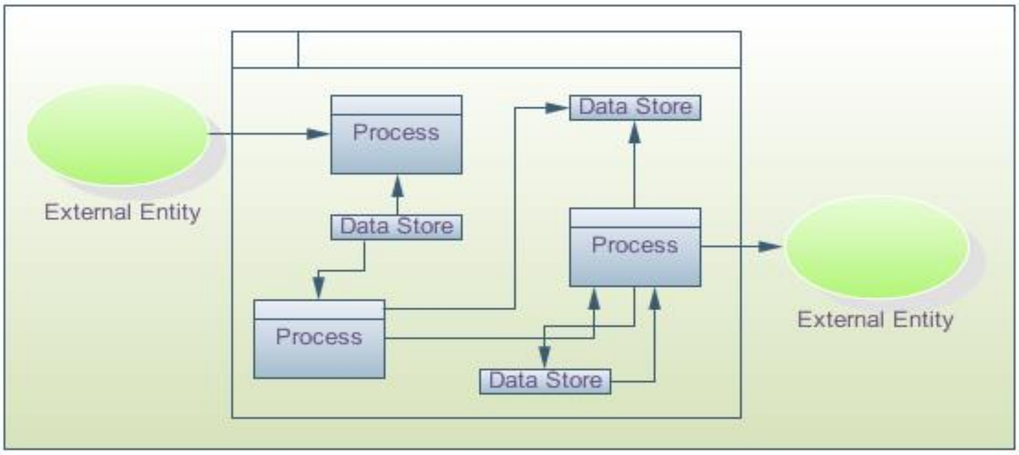
4.2 Data Flow Diagram

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel unlike a flowchart which also shows this information.

It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

4.2.1 Data Flow Diagram

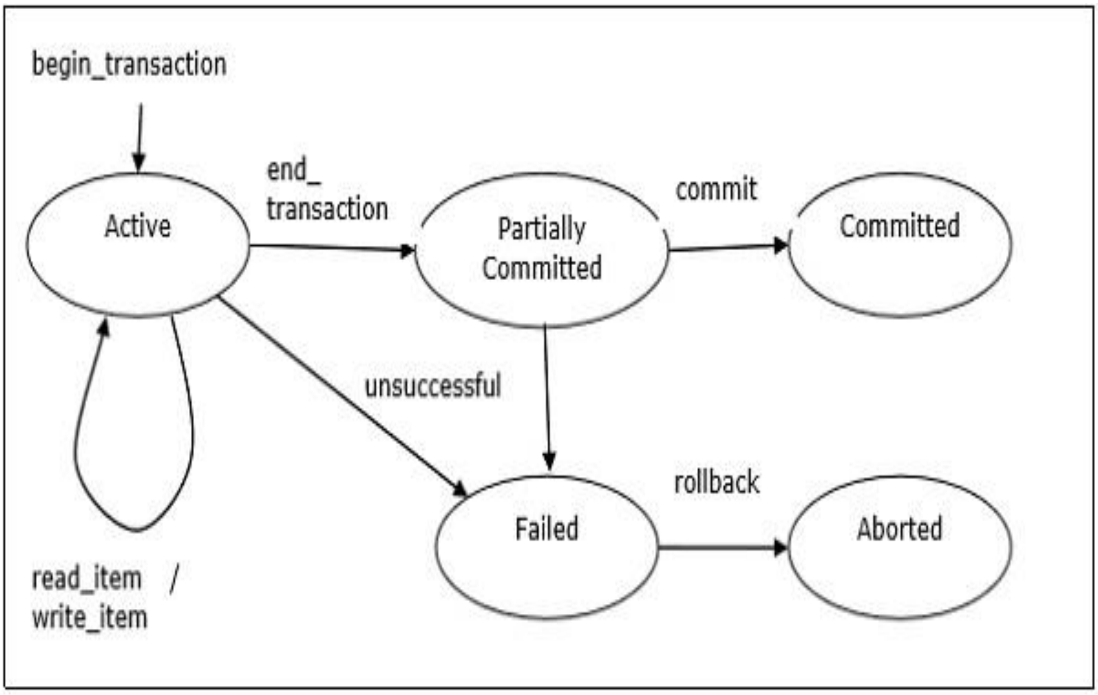


4.2.1 Top level Data Flow Diagram

4.3 State Transition Diagram

An STD is a way of describing the time-dependent behaviour of a system. The basic consistency rule is: "A system's behaviour in any state must be the same, no matter by which path the state is arrived at".

State diagrams are used to give an abstract description of the behaviour of a system. This behaviour is analysed and represented as a series of events that can occur in one or more possible states. Hereby "each diagram usually represents objects of a single class and track the different states of its objects through the system".



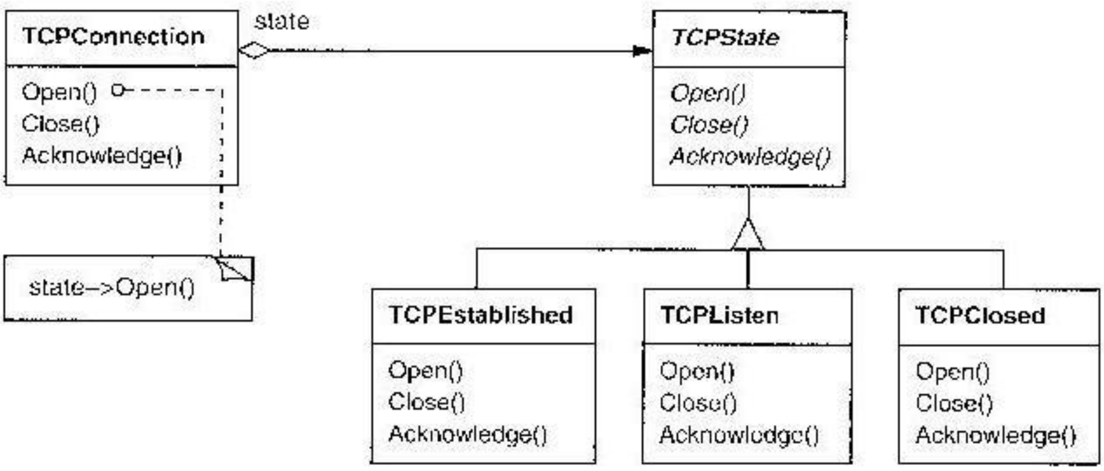
4.3 State Transition Diagram

4.4 Collaboration Diagrams:

A collaboration diagram, likewise called a correspondence graph or communication outline (graph), is a delineation of the connections and associations among programming (s/w) objects in the Unified Modelling Language (UML).

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

4.4.1 User's Collaboration Diagram



4.4.1 User's collaboration diagram

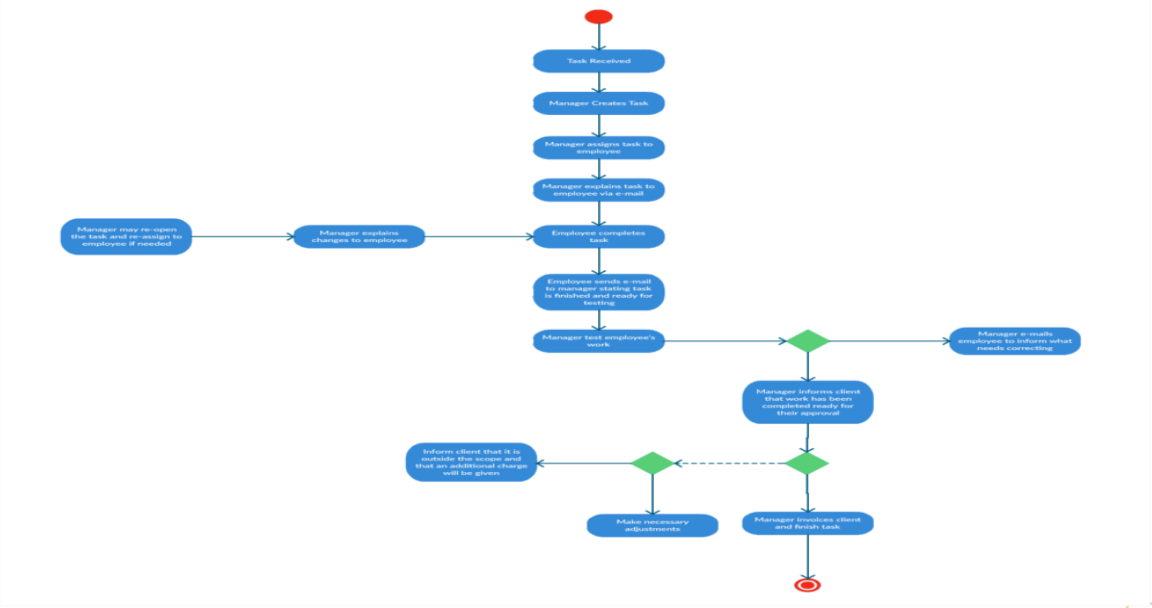
4.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial node) of the workflow;
- An encircled black circle represents the end (final node).

4.5.1 User Activity Diagram:


Task Received
Manager Creates Task
Manager assigns task to employee
Manager explains task to employee via e-mail
Manager may re-open the task and re-assign to employee if needed
Manager explains changes to employee
Employee completes task
Employee sends e-mail to manager stating task is finished and ready for testing
Manager test employee's work
Manager e-mails employee to inform what needs correcting
Manager informs client that work has been completed ready for their approval
Inform client that it is outside the scope and that an additional charge will be given
Make necessary adjustments
Manager invoices client and finish task

4.5.1 User activity Diagram

5. CHANGE MANAGEMENT PROCESS

Change Management is the process of planning, coordinating, implementing and monitoring changes affecting any production platform within application’s control. The main objectives of the Change Management process in my application are to:

- ✓ Ensure that changes are made with minimum disruption to the services IT has committed to its users.
- ✓ Support the efficient and prompt handling of all changes.
- ✓ Provide accurate and timely information about all changes.
- ✓ Ensure all changes are consistent with business and technical plans and strategies.
- ✓ Ensure that a consistent approach is used.
- ✓ Provide additional functionality and performance enhancements to systems while maintaining an acceptable level of user services.
- ✓ Reduce the ratio of changes that need to be backed out of the system due to inadequate preparation.
- ✓ Ensure that the required level of technical and management accountability is maintained for every change.
- ✓ Monitor the number, reason, type, and associated risk of the changes.

A. Appendices

This software will help in maintaining the information regarding each and every project which may be running currently or it may be delivered project. The main purpose of this software is to computerize the whole process of maintaining the information and transaction records of the projects.

- ✓ JAVA2 The Complete Reference - Herbert Schildt
- ✓ An Introduction to HTML – J.Arnold
- ✓ Mastering JavaScript - James Jaworski
- ✓ HTML 3.2 UNLEASED - John December and Mark Gins
- ✓ JAVA handbook - TATA InfoTech
- ✓ Credit Card Banking - P.C. Nanda