

PROJECT REPORT ON

V BUILD

Submitted in partial fulfillment of the requirement for the award of degree in

MASTER OF COMPUTER APPLICATIONS

of the

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted by

FATHIMA THASLIYA PA (NCE20MCA-2022)

Under the guidance of

Dr.DEEPA A,Associate Professor, Dept. of MCA



**DEPARTMENT OF MCA NEHRU COLLEGE OF ENGINEERING
AND RESEARCH CENTRE,**

(NAAC Re-Accredited with “A” grade)

PAMPADY, THIRUVILWAMALA, THRISSUR-680567 JULY 2022

PROJECT REPORT ON

V BUILD

Submitted in partial fulfillment of the requirement for the award of degree in

MASTER OF COMPUTER APPLICATIONS

of the

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



Submitted by

FATHIMA THASLIYA PA (NCE20MCA-2022) Semester 4 MCA (2022-23)

Under the guidance of

Dr.DEEPA A,Associate Professor, Dept. of MCA



**DEPARTMENT OF MCA NEHRU COLLEGE OF ENGINEERING
AND RESEARCH CENTRE, JULY 2022**



NEHRU COLLEGE OF ENGINEERING AND RESEARCH

CENTRE

DEPARTMENT OF MCA

COLLEGE VISION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

COLLEGE MISSION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values. We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

DEPARTMENT VISION

To create a school of distinction for the PG students, prepare them to be industry - ready, and achieve Academic excellence by continuous endorsement of the Faculty team in terms of Academics, Applications & Research.

DEPARTMENT MISSION

The Department of Computer Applications strives to provide quality and competency-based education and fine-tune the younger generation through Curricular, Co-Curricular and Extracurricular activities so as to encounter the Professional and Personnel challenges ahead with

Pragmatic skills & courage , thereby 'Creating the True Citizens'.

DECLARATION

I hereby declare that the project report entitle’’**V BUILD**’’ Submitted to the Department of MCA at Nehru College Of Engineering And Research Centre in partial fulfillment of the requirement for the award of degree in **MASTER OF COMPUTER APPLICATIONS from APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**, is a record of original work done by me under the guidance of Mrs. DIVYA P, Assistant Professor of Department of MCA, during my Fourth Semester MCA course period 2022.

FATHIMA THASLIYA P A

**NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE,
PAMPADY**



CERTIFICATE

This is to certify that, the work entitled “**V BUILD**” has been presented by FATHIMA THASLIYA P A, **NCE20MCA-2022** of Fourth Semester MCA in Partial Fulfillment of the requirement for the award degree **MASTER OF COMPUTER APPLICATIONS, APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY.**

We also certify that the work done is original.

Project Guide

Head of the Department

Date

Principal

ACKNOWLEDGEMENT

First and most, I thank the **God Almighty** for showing me the path to the completion of project work. I thank **Prof. Dr. Karibasappa Kwadiki**, the principle **NCERC**, for providing good atmosphere for project completion and presentation. I thank **Dr. Sudheer S Marar**, Head of MCA Department, NCERC for his valuable suggestions and support throughout the project work. I wish to acknowledge my deep sense of gratitude and heartfelt thanks to My Project Guide **Dr.DEEPA A**Assistant professor, MCA Department for her valuable suggestions, precious time, kind hearted motivations throughout the project work period. I express my immense gratitude to all my friends, without whom I would have never been able to do my project well.

Abstract

The project work entitled, “**V BUILD**” is a website with a simple user-friendly interface. The system is built with Python as the front end and uses SQLite as the back end. The user interfaces are using CSS, HTML, JavaScript and Jinja.

In the modern era, everyone is interested in creating or modifying their home, shop or any type of building in a modernized and an attractive manner. The mobilization of all factors of production for this purpose makes the building construction quite time consuming. Though this project aims at bringing together all requirements of the construction of building and mobilization of all factors of production such as labor, material etc. The purpose of this project is to assist the general public who are interested to build their concepts in an attractive manner.

Contents

Declaration	i
Certificate	ii
Acknowledgements	iii
Abstract	iv
Contents	v
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Background	2
1.1.1 Motivation	2
1.2 Objective	2
1.3 Contribution	3
1.4 Report Organization	3
2 Literature Survey	4
3 Methodology	6
3.1 Introduction	6
3.1.1 Modules And Descriptions	6
3.2 Workflow	7
3.3 Implementation	14
4 Agile Methodology	15
4.1 Introduction	15
4.2 User Story	17

Contents

Project Plan	17
Sprint Backlog (Plan)	19
Sprint Backlog (Actual)	24
Product Backlog Review	28
Sprint Review	30
Testing and Validation	32
Git	33
5 Result and Discussions	34
6 Conclusion	44
References	45
Appendix	47
Source Code	47
Git	
History.....	51

List of Figures

3.1	Workflow Diagram	7
	A. 9 Results	50
	A. 10 Git Graph 1	51

List of Tables

User Story	16
Product Backlog	17
Project plan	18
Sprint Backlog (Plan) - Sprint 1	21
Sprint Backlog (Plan) - Sprint 2	22
Sprint Backlog (Plan) - Sprint 3	23
Sprint Backlog (Plan) - Sprint 1	25
Sprint Backlog (Plan) - Sprint 2	26
Sprint Backlog (Plan) - Sprint 3	27
Product Backlog Review (Sprint1)	28
Product Backlog Review (Sprint2)	28
Product Backlog Review (Sprint3)	29
Sprint Review (Sprint1)	26
Sprint Review (Sprint2)	27
Sprint Review (Sprint3).....	28
Testing and Validation(Sprint1)	28
Testing and Validation(Sprint2)	29
Testing and Validation(Sprint3).....	30

Chapter 1

Introduction

1.1 Background

In the modern era, everyone is interested in creating or modifying their home, shop or any type of building in a modernized and an attractive manner. The mobilization of all factors of production for this purpose makes the building construction quite time consuming. Though this project aims at bringing together all requirements of the construction of building and mobilization of all factors of production such as labour, material etc. The purpose of this project is to assist the general public who are interested to build their concepts in an attractive manner.

1.1.1 Motivation

During the system analysis phase, the system is analyzed such that V BUILD consists of admin, consumer, product supplier and service provider. The existing system is applications that provide an estimate of total cost for the construction and applications that provide the details of raw materials. There is no application or site that is available all together like products, services and real estate. To overcome the difficulties of the existing system, the proposed system called V BUILD is developed. The proposed system is an efficient, easy to manage, user friendly python application.

1.2 Objective

This project aims at developing an efficient and user-friendly python application. It also saves time and provide an easy way to get information about products, services and real estate.

- 2 The main objective of the system is to manage and control the progress of construction projects.
 - 3 We can book services and products from the website.
 - 4 To reduce time and effort.
 - 5 The system is very helpful for people.
-

1.3 Contribution

The major contribution in this project are:

1. Designed and developed a new system for providing construction materials.

1.4 Report Organization

The project report is divided into six sections. Section 2 describes literature survey. Section 3 describes the methodology and section 4 describes agile methodology used for implementing The project. Section 5 gives the result and discussions. Finally section 6 gives the conclusion.

Chapter 2

Literature Survey

The focus on the site is to save the user's time when it comes to purchase any material and wants any service. The construction industry is fragmented with many stakeholders in construction project. We got the idea about the management of construction materials. Also the idea for planning the construction services procedure. From the base paper we got the idea for all the requirement of building and user while constructing. So we added the materials and services as per requirement of user. Material management can be define as an organizing functions responsible for planning as well as controlling the materials flow. Most company or organization are having the same problem related to the managing the materials. Besides that, the web application provides instruction to the user to more understand to use the web application. Furthermore, they want to make user any experience as safe and comfortable as possible. We offer a wide choice of any service and all material are well maintained in the particular and in perfect way.

Stephen Sewalk*, J.Mark Taylor*, Paul Chinowsky "A Survey of Construction Management Programs: Publication, Expectations and Compensation" [2] In this paper to –date no comprehensive study related to construction management systems. The construction is essential industry in every country for significant component of any countries. The society depends on construction industry to build good and residential house. This paper deals with the management of construction materials while constructing buildings. This approach used to manage the materials for infrastructure. Great deal to use the construction materials prominently so that its wastage should be avoided. Wendy L. Tate*, Lisa M. Ellram*, Jon F.Kirchoff "Corporate Social Responsibility Reports: A Thematic Analysis Related To Supply Chain Management" [3]. In this paper the scale and scope of human generated activity has put the pressure on the natural systems, companies, governments and citizens have concern about being of society and environment. Antti Ruuska*, Tarja Hakkinen* "Material Efficiency

of Building Construction” [4]. In this article the resource efficiency i.e. use of natural resources and materials, in order to create the construction products in natural way with the lesser resources and for good environmental impact. As the article deals with efficiency and the use of construction material. Also, flow to manage the construction materials with efficiency without its impact on environment. In this the materials are classified be used more conveniently whenever requirement is needed. Chris I. Goodier*, Raman Mangabhai*, Mark Tyrer*, Edwin Trout “The Future of Construction Materials Research” [5]. In this paper communication between youngsters and materials researchers. In this, revealed that there must be communication between sealers and purchases. So that they can speak their requirement. Rakeshnayak*, Mukeshpandey* “Management of Construction Materials on Project Site” [6] In this paper give the idea about the management of the construction materials required for the project. Which is involved in this paper is planning, execution and evaluation of all type of construction work. We give the idea about from this paper is purpose of material management i.e. to assure that the right material are the right place, in good quantity when needed. Jose Ignacio Ortiz-Gonzalez*, Eugenio Pellicer*, Gregory Howell* “Contingency Management In Construction Projects: A Survey Of Spanish Contractors” [7] From this paper we got an idea about the how contingency management in the construction based projects. In this paper the main focus on the risk of delivery of any type of the construction material. Also, in this paper show the uncertainty related to the contractor. In this paper also proved the construction companies usually set time and cost contingencies with the goal of the construction management. The risk and uncertainty are present in all constructions projects.

Chapter 3

Methodology

3.1 Introduction

V BUILD is Python web based application which overcome all the drawbacks of existing system and provides an efficient and responsive web application for getting the details about booking, buying of construction products and services. Proposed system consists of mainly four modules, they are:

- ✚ Admin
- ✚ Consumer
- ✚ Product supplier
- ✚ Service provider

3.2 MODULES AND DESCRIPTIONS

ADMIN

Admin manages the system. All activities in the system is controlled by admin.

- ✚ Login
- ✚ View Profile
- ✚ View Users
- ✚ Approve Users
- ✚ View Products

- ✚ Approve Products
- ✚ View Services
- ✚ Approve Services
- ✚ View real estate
- ✚ Approve real estate
- ✚ Manage Payment
- ✚ Manage Ticket
- ✚ View Feedback

CONSUMER

Consumer is the user who can buy products and services and also add their real estate and view others real estate.

- ✚ Register
- ✚ Login
- ✚ View Profile

- ✚ View Products
- ✚ View Product Category
- ✚ View Services
- ✚ View Service Category
- ✚ Add real estate
- ✚ View real estate
- ✚ Manage Product booking
- ✚ Manage Service booking
- ✚ Manage Payment
- ✚ Send Ticket
- ✚ Send Feedback

PRODUCT SUPPLIER

Product Supplier is the user who can add their products.

- ✚ Register
- ✚ Login

-
- ✚ View Profile
 - ✚ Manage Product details
 - ✚ View booking details
 - ✚ View Payment
 - ✚ Send Ticket
 - ✚ View Feedback

SERVICE PROVIDER

Service Provider is the user who can add their services.

- ✚ Register
- ✚ Login
- ✚ View Profile
- ✚ Manage service details
- ✚ View booking details
- ✚ View Payment
- ✚ Send Ticket
- ✚ View Feedback

3.3 Workflow

Fig.3.3 : Workflow Diagram

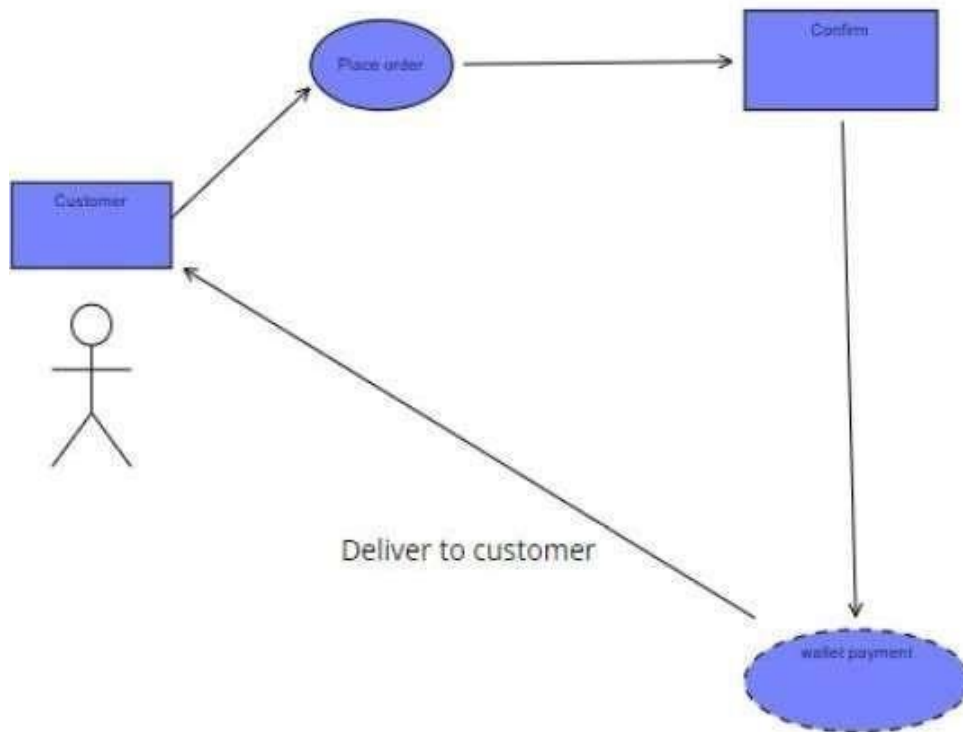


Fig.3.3 : Workflow Diagram

3.4 Software Environment

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and crossplatform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

-
- It supports functional and structured programming methods as well as OOP.
 - It can be used as a scripting language or can be compiled to byte-code for building large applications.
 - It provides very high-level dynamic data types and supports dynamic type checking.
 - It supports automatic garbage collection.
 - It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information.
```

```
>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ('Hello, Python!')**; However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active. Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.
```

This produces the following result –

```
Hello, Python!
```

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.

-
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.
- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files. Python Syntax compared to other programming languages
- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe): `C:\Users\Your Name>python --version`

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: <https://www.python.org/>

3.4 Implementation

Implementation Platform and Language

Python is a general purpose, interpreted-high level language used extensively nowadays for solving domain problems instead of dealing with complexities of a system. It is also termed as the ‘batteries included language’ for programming. It has various libraries used for scientific purposes and inquiries along with number of third-party libraries for making problem solving efficient.

In this work, the Python libraries of Numpy, for scientific computation, and Matplotlib, for 2D plotting have been used. Along with this, Pandas tool of Python has been employed for carrying out data analysis. As a development platform, Pycharm, which proves to work great due to its excellence in ‘literate programming’, where human friendly code is punctuated within code blocks, has been used.

Chapter 4

Agile Methodology

4.1 Introduction

After the initial studies it is found that agile model of software development is suitable and is the best method for the development of this system. Agile methodology mainly focused on the client satisfaction through continuous delivery. Also it sets a minimum number of requirements and turns them in to a deliverable product. As this project has many individual requirements which can be delivered in parts and the user can gradually improve their work efficiency. Agile methodology has a family of methods of which scrum is selected for the development of this project. Scrum is process framework that has been used to manage complex product development. It is not a process or technique for building products rather it is a framework within which various processes can be employed. Also it is suitable method to support the development process. It focuses on lean software development and has in building better software effectively and efficiently.

Agile is one of the most widely used and recognized software development framework. The methodology those experts agreed upon was described as ‘lightweight’ and fast. Agile is also about being the adaptive and continuous improvement, as much as it is about constant feedback and speed of delivery.

Agile is a software development approach where a self-sufficient and cross-functional team works on making continuous deliveries through iterations and evolves throughout the process by gathering feedback from the end users.

The major rules in scrum methodology are:

1. **The product owner (PO)** : Who represents the stakeholder and the business.

2. **The scrum master** : Ensures the process followed, removes obstructions, and protects the development system
3. **Development team**: Cross functional, self organizing team who actually do the actual analysis, design implementation and testing process.

They work together in iterative time boxed durations called sprints. The first step is the creation of the product backlog by the PO. It's a to-do list of stuff to be done by the scrum team. Then the scrum team selects the top priority items and tries to finish them within the time box called a sprint. An easier way to remember all of this is to memorize the 3-3-5 frame- work. It means that a scrum project has 3 roles, 3 artifacts, and 5 events

These are:-

1. **Roles**: Product Owner, Scrum Master, and development team.
2. **Artifacts**: Product Backlog, Sprint Backlog and Product Increment.
3. **Events**: Sprint, Sprint planning, Daily Scrum, Sprint review and Sprint retrospective

The framework begins with a simple premise start with what can be seen or known. After that the progress is tracked and tweak as necessary. The three pillars of scrum are transparency, inspection and adaptation. In scrum everyone has a role.

The Git is used as the version control system for this project. Version control is a system that records changes to a file or set of files over time so that a specific versions can be recalled later. Version control systems are a category of software tools that help a software team for managing changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

4.2 User Story

A user story is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement.

Us er Story ID	As a <Type of user>	I want to perform <some task>	So that I can <achieve some goal>
1	Product suppliers , service providers & Consumer	Register to the system	Access the system
2	Product suppliers , service providers , Consumer & Admin	Login	Access the account
3	Product suppliers	Add construction materials	Store construction material details with pricing
4	Consumer	Search	Construction material availability & pricing
5	Consumer	Order	To buy Construction material
6	Product suppliers	Deliver	Deliver Construction material to consumer

Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome. The product backlog is the single authoritative source for things that a team works on.

That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.

It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories. The product backlog of the system is given in Table 4.2

PRODUCT BACKLOG			
ID	Name	Priority	Estimate [Hrs]
1	Registration	1	40
2	Login	2	40
3	Add construction materials	3	24
4	Search	4	24
5	Order	5	32
6	Deliver	6	32
7	Reports	7	32

Table 4.2 product Backlog

4.3 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project plan is given in Table 4.3

User Story ID	Task name	Start date	End date	Days	Status (to be filled by scrum master)
Sprint 1		23/02/2023	08/03/2023	14	Completed
1	Registration	23/02/2023	24/02/2023	2	Completed
1	Coding	27/02/2023	03/03/2023	5	Completed
1	Testing	06/03/2023	08/03/2023	3	Completed
Sprint 2		09/03/2023	22/03/2023	14	Completed
1	Coding	09/03/2023	17/03/2023	7	Completed
1	Testing	20/03/2023	22/03/2023	3	Completed
Sprint 3		23/03/2023	05/04/2023	14	Completed
1	Database connectivity	23/03/2023	03/04/2023	8	Completed
1	Testing	04/04/2023	05/04/2023	2	Completed
Sprint 4		06/04/2023	19/04/2023	14	Completed
1	Deployment	06/04/2023	12/04/2023	5	Completed
1	Testing and validation	13/04/2023	19/04/2023	5	Completed

Table 4.4 Project Plan

The Project has three sprints:

1. Sprint 1

Two tasks are planned in this sprint. First one is load pre-trained models, initial coding and testing.

2. Sprint 2

Three tasks are planned in this sprint. First one is design and impute missing values, modelling and next one is testing.

3. Sprint 3

In this sprint two tasks are planned to complete, one is Deployment of prediction system and second is testing and result discussion.

4.5 Sprint Backlog (Plan)

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some

number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete.

Sprint 1:

Two tasks are planned in this sprint. First one is load pre-trained models, initial coding and testing.

Sprint 2:

Three tasks are planned in this sprint. First one is design and impute missing values, modelling, coding and next one is testing.

Sprint 3:

In this sprint two tasks are planned to complete, one coding and report generation

Sprint 4:

In this sprint there is 2 task deployment & validation given in table 4.7

Backlog item	Completion date	Original estimate in hrs.	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12
User story#1hrs.			Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.

Chapter 4. *Agile Methodology*

Registration	24/02/2023	6	1	1	1	1	1	1	0	0	0	0	0	0
Coding	03/03/2023	4	1	1	1	1	0	0	0	0	0	0	0	0
Testing	08/03/2023	4	0	0	0	0	0	0	0	0	1	1	1	1
Total		14	2	2	2	2	1	1	0	0	1	1	1	1

Table 4.51: Sprint backlog(plan)-sprint1

Backlog item	Completion date	Original estimate in hrs.	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12
User story#1 hrs.			Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.
Coding	17/03/2023	6	1	1	1	1	1	1	0	0	0	0	0	0
Testing	22/03/2023	5	0	0	0	0	0	0	0	1	1	1	1	1
Total		11	1	1	1	1	1	1	0	1	1	1	1	1

Table 4.52: Sprint backlog (plan) - sprint 2

Backlog item	Completion date	Original estimate in hrs.	Day 1	Day 2	Day 3	Day 4
User story#1			Hrs.	Hrs.	Hrs.	Hrs.
DB connectivity	03/04/2023	5	2	1	1	1
Testing	05/04/2023	7	2	1	2	2

Total		12	4	2	3	3
-------	--	----	---	---	---	---

Table 4.53: sprint backlog (plan) – sprint 3

Backlog item	Completion date	Original estimate hrs.	Day 1	Day 2	Day 3
User story#1			Hrs.	Hrs.	Hrs.
Deployment	12/04/2023	4	2	1	1
Testing and validation	19/04/2023	4	2	1	1
Total		8	4	2	2

Table 4.54:
sprint backlog
(plan) – sprint

Backlog item	Completion date	Original estimate in hrs.	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12
Chapter 4. Agile Methodology														
User story#1hrs.			Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.
Registration	24/02/2023	6	1	1	1	1	1	1	0	0	0	0	0	0
Coding	03/03/2023	4	1	1	1	1	0	0	0	0	0	0	0	0
Testing	08/03/2023	4	0	0	0	0	0	0	0	0	1	1	1	1
Total		14	2	2	2	2	1	1	0	0	1	1	1	1

4.4 Sprint Backlog (Actual)

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below.

Backlog item	Completion date	Original estimate in hrs.	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12
User story#1hrs.			Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.	Hrs.
Coding	17/03/2023	6	1	1	1	1	1	1	0	0	0	0	0	0
Testing	22/03/2023	5	0	0	0	0	0	0	0	1	1	1	1	1
Total		11	1	1	1	1	1	1	0	1	1	1	1	1

Table 4.62: Sprint Backlog (actual) – sprint 2

Backlog item	Completion date	Original estimate in hrs.	Day 1	Day 2	Day 3	Day 4
User story#1			Hrs.	Hrs.	Hrs.	Hrs.
DB connectivity	03/04/2023	5	2	1	1	1
Testing	05/04/2023	7	2	1	2	2
Total		12	4	2	3	3

Table 4.63: Sprint backlog (actual) – sprint 3

Backlog item	Completion date	Original estimate in hrs.	Day 1	Day 2	Day 3
User story#1			Hrs.	Hrs.	Hrs.
Deployment	12/04/2023	4	2	1	1
Testing and validation	19/04/2023	4	2	1	1
Total		8	4	2	2

Table 4.64: Sprint Backlog(actual) -Sprint 4

Product Backlog Review

REVIEW FORM

SPRINT 1

Version:1.0

Date:18/08/2022

User story ID	Comments from scrum master if any	Comments from product owner if any
1	Developer should have a easy registration process	User friendly registration

2	Effective login	If there is forgot password or user name handled
---	-----------------	--

Table 4.71: Product Backlog Review (Sprint 1)

SPRINT 2**Version :1.0****Date:15/09/2022**

User story ID	Comments from scrum master if any	Comments from product owner if any
3	Should check the data inserted correctly	Inserted
4	Design and development of registration forms	Should customize different models

Table 4.72: Product Backlog Review (sprint 2)

SPRINT 3**Date:16/10/2022****Version:1.0**

User story ID	Comments from scrum master if any	Comments from product owner if any
5	Should check database connectivity	Check connection

Table 4.73:Product Backlog Review (sprint 3)

SPRINT 4**Version: 1.0****Date:05/11/2022**

User story ID	Comments from scrum master if any	Comments from product owner if any

6	Deployment	Visualize final output
7	Generate predicted result	satisfied

Table 4.74: Product Backlog Review (Sprint 4)

4.8 Sprint Review

At the end of each sprint a Sprint Review meeting is held. During this meeting the Scrum Team shows which Scrum Product Backlog items they completed (according to the Definition of Done) during the sprint. This might take place in the form of a demo of the new features. Backlog items that are not completed shall not be demonstrated. Otherwise, this might suggest that these items are finished as well. Instead, incomplete items/remaining activities shall be taken back into the Scrum Product Backlog, re-estimated and completed in one of the following sprints. The Sprint Review meeting should be kept very informal. No PowerPoint slides should be used and time for preparation and performing the meeting should be limited. During the meeting the Scrum Product Owner inspects the implemented backlog entries and accepts the solution or adds new stories to the Scrum Product Backlog to adapt the functionality. Participants in the sprint review typically include the Scrum Product Owner, the Scrum Team and the Scrum Master. Additionally, management, customers, and developers from other projects might participate as well.

REVIEW FORM

SPRINT 1

Version: 1.0**Date:18/08/2022**

User story ID	Comments from scrum master if any	Comments from product owner if any
1	Developer should have a easy registration process	Satisfied
2	Effective login	successful

Table 4.81: Sprint Review (Sprint 1)

*SPRINT 2***Version: 1.0****Date:15/09/2022**

User story ID	Comments from scrum master if any	Comments from product owner if any
3	Should check data inserted correctly	successful
4	Effective login	Satisfied

Table 4.82: Sprint Review (Sprint 2)

*SPRINT 3***Version: 1.0****Date:16/10/2022**

User story ID	Comments from scrum master if any	Comments from product owner if any
5	Should check database connectivity	Connection successful

Table 4.83: Sprint Review (Sprint 3)

SPRINT 4

Version: 1.0**Date:05/11/2022**

User story ID	Comments from scrum master if any	Comments from product owner if any
7	Deployment completed	Satisfied
8	Output generated	Satisfied with result

Table 4.84: Sprint Review (Sprint 4)

4.7 Testing and Validation

Sprint 1

Test #	Date	Action	Expected result	Actual Result	Result Pass? (Yes/No)
1	24/02/2023	Registration	Registration successful	successful	yes
2	08/03/2023	Login	Login to system	Login successful	yes

Table 4.91: Testing and validation (Sprint 1)

Sprint 2

Test#	Date	Action	Expected result	Actual result	Result pass? (Yes/No)
-------	------	--------	-----------------	---------------	-----------------------

Chapter 4. Agile Methodology

1	22/03/2023	Development if transfer learning models	Can choose the best model	Done	yes
---	------------	---	---------------------------	------	-----

Table 4.92: Testing and validation (Sprint 2)

Sprint 3

Test#	Date	Action	Expected result	Actual result	Result pass? (Yes/No)
1	05/04/2023	Development of website	UI will be formed	Done	yes

Table 4.93: Testing and validation (Sprint 3)

Sprint 4

Test#	Date	Action	Expected result	Actual result	Result pass? (Yes/No)
1	12/04/2023	Deployment	Get doctor consultation	Done	yes

Table 4.94: Testing and validation (Sprint 4)

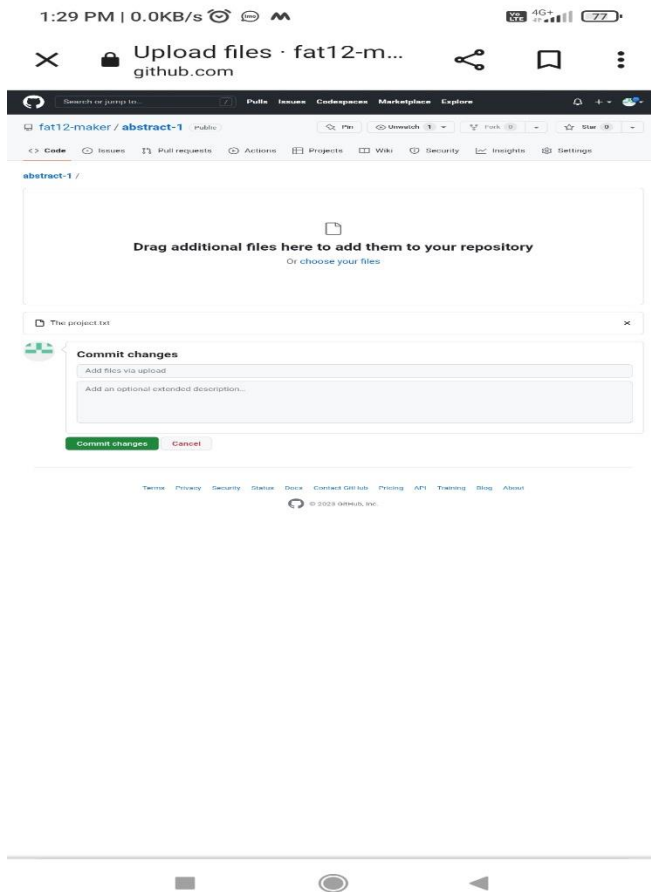
4.8 Git

The Git is used as the version control system for this project. Version control is a system that records changes to a file or set of files over time so that a specific versions can be recalled later. Version control systems are a category of software tools that help a software team for managing changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. To show the continuous

development of the project the Gitlab histories are shown in below;

Chapter 4. Agile Methodology



Chapter 5

Result and Discussions

The project has a very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database, the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

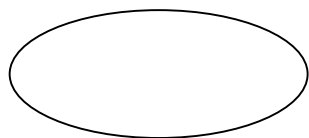
- ❖ More shops can be accommodated for the ordering system.
- ❖ Mobile app for tracking the system's scenario.
- ❖ Mobile text alerts for communicate faster with an SMS service.
- ❖ Corresponding website links will be provided with the schemes.

System Design

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

Symbols



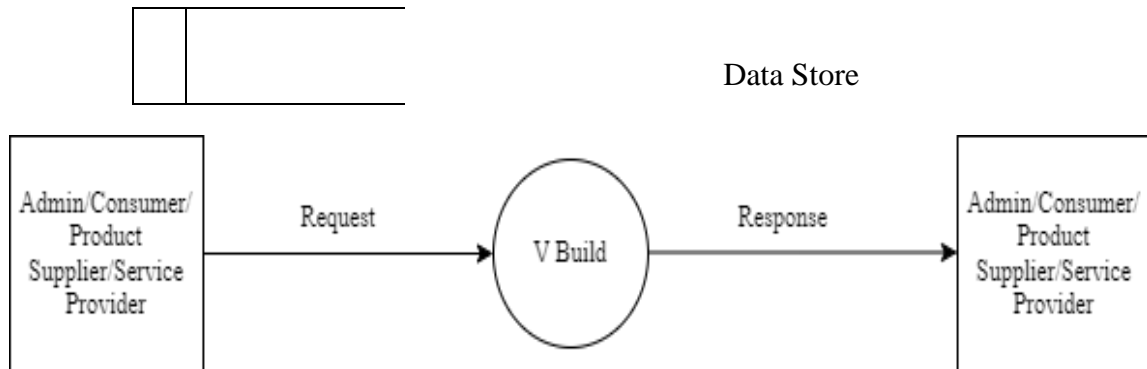
Process



Elements reference

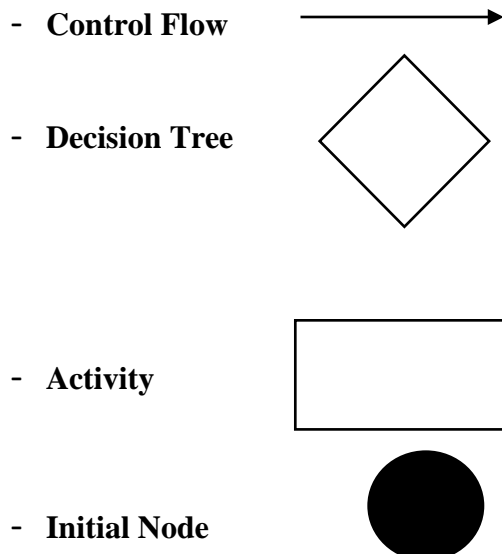
Data Flow Process

Source link



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 can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



- **End Node**



- **Fork**



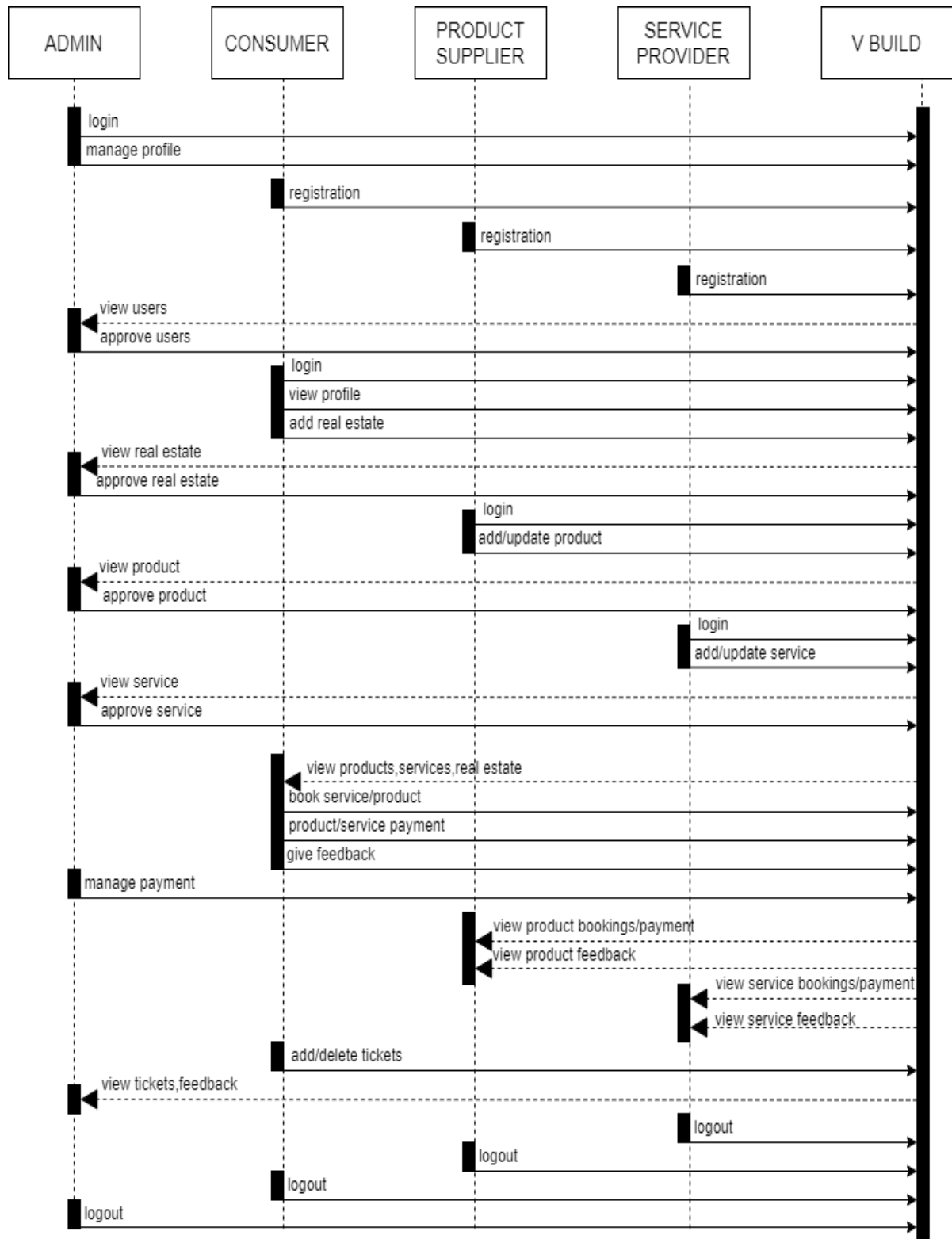
- **Join**

Fork is a black bar with one flow going into it and several leaving it. This denotes the beginning of parallel activity. Join is a black bar with several flows entering it and one leaving it. All flows going into the join must reach it before processing may continue. This denotes the end of parallel processing.

Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. Messages written on horizontal arrows and the message name written above them, it displays interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous

Sequence Diagram



SYSTEM REQUIREMENTS

Hardware Specification

Processor	: Pentium IV 2GHz and above
RAM	: 2 GB
Hard disk drive	: 500 GB
Monitor	: 15" color monitor
Key board	: 102 keys
Mouse	: Scroll Mouse

3.2 Software Specification

Operating system	: Windows 7 or higher
Web Browser	: Mozilla Firefox, Google Chrome or later
Front end	: HTML, JavaScript, CSS, Jinja
Back end	: Python, SQLite3
Framework	: Django
IDE	: Microsoft Vscode
Documentation	: Microsoft Word 2010

5. SYSTEM TESTING

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system.

- ✦ Unit testing
- ✦ Integration testing
- ✦ Validation testing
- ✦ User acceptance testing.

5.1 Unit Testing

It focuses effort on the smallest unit of software designs in the module. This is also known as “Module Testing”. To check whether each module in the software is proper and it gives desired outputs to the given inputs. All validations and conditions are tested in the module in the first unit. Control paths are tested to ensure the information properly flows into, and out of the program unit under test. Boundary conditions are tested to ensure that they operate at boundaries. The proposed system V Build has four different modules and hence each module is coded and tested differently. Since the admin module requires more security it should be tested carefully. In order to provide strong security or administrator view sessions are added in the code. So, testing has to be done for each activity of admin. The other users of V Build are Admin and Users. Testing has to be done while each person login in to the system so that no one can use system with different username and password. The main objective of the application is to provide an interactive space for the users and hence each of the functions added in the platform has to be tested and need to check whether the database is linked correctly.

5.2 Integration Testing

The modules are integrated to form complete software package. It addresses the issues associated with given problem of verification and program construction. Test that part of the system at some level work together correctly the purpose of integration is to verify functional, performance and reliability requirements placed on major design items. This testing is conducted on the basis of modules. The integration testing is performed to detect design errors by focusing on testing the interconnection between modules. The objective is to take the unit tested modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complete the isolation. Here We Integrate the V Build system into a complete website. System Testing When a system is developed, it is hoped that it performs properly. In practice, however some errors always occur. The main purpose of

testing an information system is to find the errors and correct them. A successful test is one, which finds an error.

The main objectives of system testing are:

- ✦ To ensure during operating the system will perform as per specification.
- ✦ To make sure that the system meets user's requirements during operation.
- ✦ To verify that the controls incorporated in the system function as intended.

5.3 Validation Testing

Validation Testing is the process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can reasonable expected by the user. That means the input we are entering to the system V Build will be checked whether the entered input is valid or not.

5.4 User Acceptance Testing

This is the key factor for the success of any system. User acceptance is done by constantly keeping in touch with prospective system at the time of development and making changes whenever required. This is done with regard to the input and output screen designs

The Different Test Cases are

- ✦ Guarantee that all independent parts within a module have been exercised at least once. In V Build the independent Parts of the system Tested and it fulfil the acceptance of the user requirements.
- ✦ Exercise all logical design on their true/false side
- ✦ Exercise all loops at their boundaries and within their operational bounds

- ✦ Each module was tested and modules were linked and integration was carried out

Levels of Testing

Client needs - Acceptance testing

Requirements - System testing

Design - Integration testing

Code - Unit testing

Testing for the Proposed System

Each module in the proposed system undergone unit testing that evaluated each module of the system. When all the forms provided exactly correct output each of them was integrated to form a particular module. Necessary validation including numeric validation, string validation, email validation and mandatory field validation were done to ensure reliability of output. Thus, V Build Care covered all the major phases of testing proves to be a system that functions according to user requirements.

Chapter 6

Conclusion

The V BUILD is developed using Python and SQLite fully meets the objectives of the system which it has been developed. The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency and all the users and blood banks associated with the system understands its advantage. The system solves the problem. It was intended to solve as requirement specification. To conclude, Project works like a component which can access all the databases and picks up different functions. It overcomes the many limitations incorporated in the present residential management system. This system has been developed to satisfy the user needs. The entire system is user friendly. The performance of the system is provided efficiently. The system wasted with all possible test data and was found to have an affective planning of the functions or processes with a high degree of accuracy and user friendliness.

FUTURE WORK:

The project has a very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database, the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

References

- Ian Somerville, Software Engineering, Pearson Education Asia, 6 th Edition
- Awad, System Analysis and Design, Calgotia, 2 nd Edition
- Elmasri, Navathe, Fundamentals of Database Systems, Pearson Education Asia,6th Edition
- <https://www.tutorialspoint.com/python>
- <https://wiki.python.org/moin/PythonBooks>
- <https://realpython.com/best-python-books>
- <https://stackify.com/learn-python-tutorials>

Appendix

Source Code

admin.py

```
from types import NoneType from
django.contrib import admin

from vbuildapp.models import Consumer, Product, ProductBooking, ProductCategory,
ProductFeedback, ProductSupplier, RealEstate, Service, ServiceBooking, ServiceCategory,
ServiceFeedback, ServiceProvider, Ticket # Register your models here. class
UsersAdmin(admin.ModelAdmin):
    list_display = ('id','name','mob','mail','approval')
search_fields = ('id','name','mob','mail','approval')
```

url.py

```
from django.urls import path
from vbuildapp import views urlpatterns
= [
    path("", views.home, name='home'),
    path('about/', views.about, name='about'),
    path('contact/', views.contact, name='contact'),
    path('logout/', views.logout, name='logout'),
    path('handlerequestservice/', views.handlerequestservice, name = 'handlerequestservice'),
    path('handlerequest/', views.handlerequest, name = 'handlerequest'),
    #consumer path('consumerregistration/', views.consumerregistration,
name='consumerregistration'), path('consumerlogin/', views.consumerlogin,
name='consumerlogin'), path('consumerhome/', views.consumerhome,
name='consumerhome'), path('consumerticket/', views.consumerticket,
name='consumerticket'), path('consumerrealestateview/', views.consumerrealestateview,
name='consumerrealestateview'), path('consumerrealestate/', views.consumerrealestate,
name='consumerrealestate'), path('consumerrealestateedit/',
views.consumerrealestateedit, name='consumerrealestateedit'),
```

```

    path('consumerservice/', views.consumerservice, name='consumerservice'),
    path('consumerservicesbooked/', views.consumerservicesbooked,
    name='consumerservicesbooked'),    path('consumerviewservice/<str:pk>',
    views.consumerviewservice, name='consumerviewservice'),
    path('consumerservicebooking/<str:pk>', views.consumerservicebooking,
    name='consumerservicebooking'),    path('consumerservicefeedback/<str:pk>',
    views.consumerservicefeedback, name='consumerservicefeedback'),
    path('consumerproduct/', views.consumerproduct, name='consumerproduct'),
    path('consumerproductsordered/', views.consumerproductsordered,
    name='consumerproductsordered'),    path('consumerviewproduct/<str:pk>',
    views.consumerviewproduct, name='consumerviewproduct'),
    path('consumerproductbooking/<str:pk>', views.consumerproductbooking,
    name='consumerproductbooking'),    path('consumerproductfeedback/<str:pk>',
    views.consumerproductfeedback, name='consumerproductfeedback'),

    #serviceproviders    path('serviceproviderregistration/',
    views.serviceproviderregistration, name='serviceproviderregistration'),
    path('serviceproviderlogin/', views.serviceproviderlogin, name='serviceproviderlogin'),
    path('serviceproviderhome/', views.serviceproviderhome, name='serviceproviderhome'),
    path('servicesadd/', views.servicesadd, name='servicesadd'),    path('servicesedit/<str:pk>',
    views.servicesedit, name='servicesedit'),    path('servicesdelete/<str:pk>',
    views.servicesdelete, name='servicesdelete'),    path('servicesview/<str:pk>',
    views.servicesview, name='servicesview'),    path('servicesbooking/<str:pk>',
    views.servicesbooking, name='servicesbooking'),    path('serviceproviderticket/',
    views.serviceproviderticket, name='serviceproviderticket'),

    #productsupplier
    path('productsupplierregistration/', views.productsupplierregistration,
    name='productsupplierregistration'),    path('productsupplierlogin/',
    views.productsupplierlogin, name='productsupplierlogin'),    path('productsupplierhome/',
    views.productsupplierhome, name='productsupplierhome'),    path('productsadd/',
    views.productsadd, name='productsadd'),    path('productsedit/<str:pk>',
    views.productsedit, name='productsedit'),    path('productsdelete/<str:pk>',
    views.productsdelete, name='productsdelete'),    path('productsview/<str:pk>',
    views.productsview, name='productsview'),    path('productsbooking/<str:pk>',
    views.productsbooking, name='productsbooking'),    path('productsuplierticket/',

```

```
views.productsuplierticket, name='productsuplierticket'), path('mails/', views.mails,
name='mails'),
]
```

emails.py

```
from django.core.mail import send_mail from
django.template.loader import render_to_string from
django.core.mail import EmailMessage from
django.core.mail import EmailMultiAlternatives
from_email = "vbuild64@gmail.com" html_template =
'mails/mails.html' def
consumer_reg(name,email,consumer): context = None
subject = "VBuild Consumer Registration"
recipient_list = [email] context =
{'consumer_reg':True,'consumer':consumer}
html_message = render_to_string(html_template, {'context':context})
msg = EmailMessage(subject, html_message, from_email, recipient_list)
msg.content_subtype = "html" # Main content is now text/html
msg.send() def sp_reg(name,email,sp):
context = None subject = "VBuild Service Provider Registration"
recipient_list = [email] context = {'sp_reg':True,'sp':sp} html_message
= render_to_string(html_template, {'context':context}) msg =
EmailMessage(subject, html_message, from_email, recipient_list)
msg.content_subtype = "html" # Main content is now text/html msg.send()
def ps_reg(name,email,ps):
context = None subject = "VBuild Product Supplier Registration"
recipient_list = [email] context = {'ps_reg':True,'ps':ps} html_message
= render_to_string(html_template, {'context':context}) msg =
EmailMessage(subject, html_message, from_email, recipient_list)
msg.content_subtype = "html" # Main content is now text/html
msg.send() #APproval def consumer_reg_approval(name,email):
subject = "VBuild Consumer Registration - Approved" message = "Hi "+name+",\nThis
is to inform you that your Consumer Account / The data you had given to update has been
approved & succesffully updated " recipient_list = [email]
```

```

send_mail(subject,message,from_email,recipient_list)
productsupplier_reg_approval(name,email):
    subject = "VBuild product supplier Registration - Approved"    message = "Hi
"+name+",\nThis is to inform you that your Product Supplier Account / The data you had
given to update has been approved & succesffully updated."    recipient_list = [email]
    send_mail(subject,message,from_email,recipient_list) def
serviceprovider_reg_approval(name,email):
    subject = "VBuild service provider Registration - Approved"
    message = "Hi "+name+",\nThis is to inform you that your Service Provider Account / The
data you had given to update has been approved & succesffully updated."    recipient_list =
[email]
    send_mail(subject,message,from_email,recipient_list) def
ticketnotify(ticketno,name,email,ticket):
    context = None    subject = "VBuild - Ticket No:
"+ticketno+" - EMail Thread"    recipient_list = [email]
    context = {'ticketnotify':True,'ticket':ticket}
    html_message = render_to_string(html_template, {'context':context})
    msg = EmailMessage(subject, html_message, from_email, recipient_list)
    msg.content_subtype = "html" # Main content is now text/html
    msg.send() def realestate_approval(rep):
    context = None    subject = "VBuild - Real Estate - Property ID:
"+str(rep.id)+". "    recipient_list = [rep.consumer.mail]    context =
{'realestate_approval':True,'rep':rep}    html_message =
render_to_string(html_template, {'context':context})    msg =
EmailMessage(subject, html_message, from_email, recipient_list)
    msg.content_subtype = "html" # Main content is now text/html
    msg.send() def product_ordering_mail(po):    context = temp =track= None
    subject = "VBuild - Product Ordered - ID: "+str(po.id)+". "    recipient_list =
[po.consumer.mail]    temp = po.AWB_no
    track = temp.split("+")    context =
{'product_booking':True,'po':po,'track':track}    html_message =
render_to_string(html_template, {'context':context})    msg =
EmailMessage(subject, html_message, from_email, recipient_list)
    msg.content_subtype = "html" # Main content is now text/html

```

```
msg.send()
def service_booking_mail(sb):
    context = temp = track = None
    subject = "VBuild - Service Booked - ID: "+str(sb.id)+". "
    recipient_list = [sb.consumer.mail]
    context = {'service_booking':True,'sb':sb}
    html_message = render_to_string(html_template, {'context':context})
    msg = EmailMessage(subject, html_message, from_email, recipient_list)
    msg.content_subtype = "html" # Main content is now text/html
    msg.send()
```

MCA@NCERC

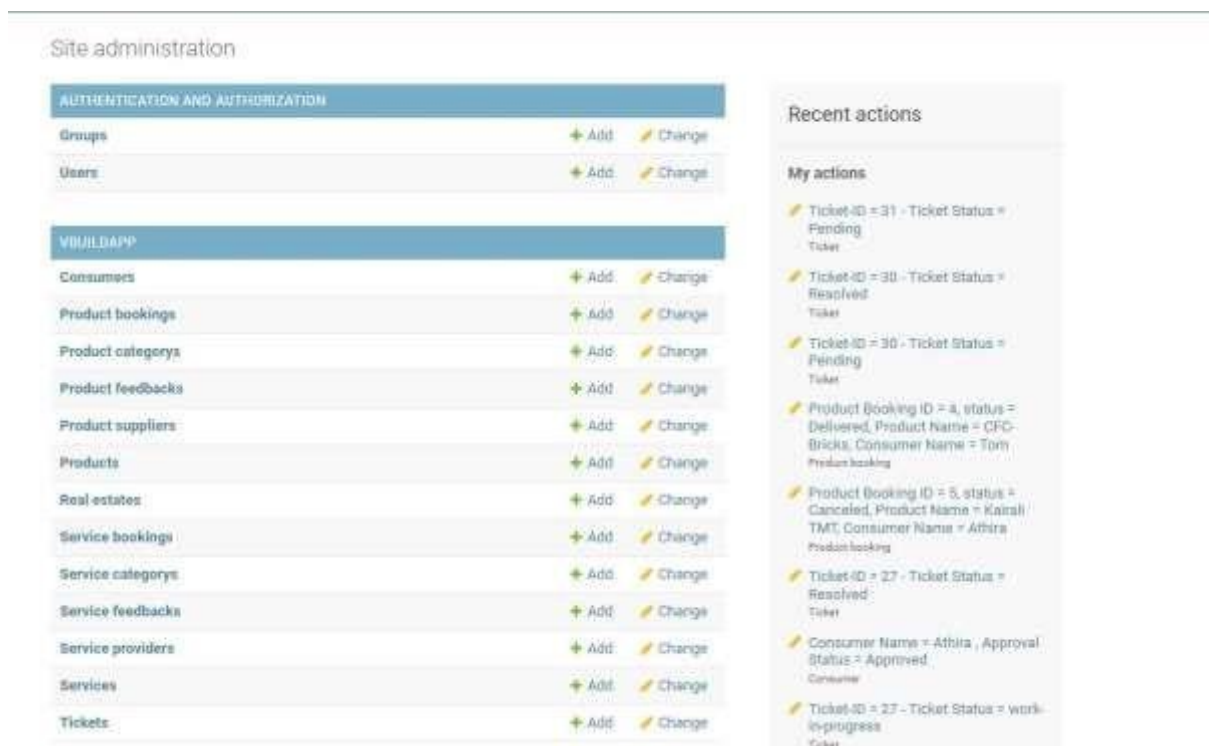
KTU

output

Home Page



Admin Page



Product Supplier Registration

**PRODUCT SUPPLIER
REGISTRATION**

Product supplier Registration

Company Name

Address

House No/Building Name

Road Name, Area, Colony

City

State

Pincode

Service Provider Registration

**SERVICE PROVIDER
REGISTRATION**

Service Provider Registration

Company Name

Address

House No/Building Name

Road Name, Area, Colony

City

State

Pincode

Consumer Registration

CONSUMER REGISTRATION

Consumer Details

Name

Address


House No/Building Name

Road Name, Area, Colony






City

State

Pincode



Consumer Login



V Build

CONSUMER LOGIN

Consumer Credentials

Email

Password

Login

Get In Touch

Quick Links

Popular Links