

COVID_19_Vaccination_Monitoring_System.pdf

Scanned on: 6:34 November 11, 2022 UTC



Identical Words	173
Words with Minor Changes	157
Paraphrased Words	103
Omitted Words	99



COVID_19_Vaccination_Monitoring_System...



Scanned on: 6:34 November 11, 2022 UTC

Results

Sources that matched your submitted document.

Software Engineering Practical File of 6th sem - SOFTWARE E https://www.studocu.com/in/document/gl-bajaj-institute-of-technology-an	4%
Uml.pptx https://www.slideshare.net/AnjaliParihar13/umlpptx	4%
SDLC - Iterative Model https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm sdlc_iterative_model.htm	3%
Unified Modeling Language (UML) Class Diagrams - Geeksf https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagr	3%
Uml Class Diagram https://www.slideshare.net/KhaledunNahar/uml-class-diagram-236999749	2%
Deployment Diagram Tutorial Lucidchart https://www.lucidchart.com/pages/uml-deployment-diagram	1%
Iterative and incremental development - Wikipedia https://en.wikipedia.org/wiki/Iterative_and_incremental_development	1%
Work Breakdown Structure (WBS) Purpose, Process and Pitf https://www.projectsmart.co.uk/work-breakdown-structure/work-breakdo work-breakdown-structure-purpose-process-pitfalls.php	1%

IDENTICAL

Identical matches are one to one exact wording in the text.

MINOR CHANGES

Nearly identical with different form, ie "slow" becomes "slowly".

PARAPHRASED

Close meaning but different words used to convey the same message.

Unsure about your report?

The results have been found after comparing your submitted text to online sources, open databases and the Copyleaks internal database. For any questions about the report contact us on support@copyleaks.com

Learn more about different kinds of plagiarism here



COVID_19_Vaccination_Monitoring_System...



Scanned on: 6:34 November 11, 2022 UTC

What Is Work Breakdown Structure, and How to Create It? https://www.invensislearning.com/blog/how-to-create-a-work-breakdown	1%
view.php?id=335106 https://elearn.daffodilvarsity.edu.bd/pluginfile.php/872108/mod_resource UML%2BInterface.docx	1%
IQBBA Exam Preparation Handbook - PDFCOFFEE.COM https://pdfcoffee.com/iqbba-exam-preparation-handbook-pdf-free.html iqbba-exam-preparation-handbook-pdf-free.html	1%
Uml class Diagram SlideShare https://es.slideshare.net/directory/slideshows/uml-class-diagram-46300557	1%
Enhancing public trust in COVID-19 vaccination: The role of https://www.oecd.org/coronavirus/policy-responses/enhancing-public-trus	0%
Apple Provided Users' Data On Government Requests https://www.cnbctv18.com/technology/apple-provided-users-data-on-gove apple-provided-users-data-on-government-requests-1785321.htm	0%
Apple transparency report: India made 27 device data, 34 fi https://tech.hindustantimes.com/tech/news/apple-transparency-report-in apple-transparency-report-india-made-27-device-data-34-financial-ide	0%
CRDT.pdf https://senior.ceng.metu.edu.tr/2017/atim/documents/CRDT.pdf CRDT.pdf	0%



COVID_19_Vaccination_Monitoring_System...



Scanned on: 6:34 November 11, 2022 UTC





COVID_19_Vaccination_Monitoring_Syste...



Scanned on: 6:34 November 11, 2022

Scanned Text

Your text is highlighted according to the matched content in the results above.

IDENTICAL MINOR CHANGES PARAPHRASED

ABSTRACT ACKNOWLEDGEMENTS **ABBREVIATIONS** INTRODUCTIONOutline: Strategy:

Stakeholder And Process Model OverviewStakeholder:

Process Model:

Identifying The Requirements System Requirement:

Software Requirement:

Functional Requirement:

Non Functional Requirement:

Work Breakdown StructureApproach:

Architecture DesignSystem Architecture:

Use Case:

Entity Relation:

Dividing into Classes:

Data Flow:

Communication Diagram:

Behavioral PatternSoftware State:

Event Scenarios:

Deployment Diagramming:

Front End:

Module ImplementationM1: Main Screen and New User Registration/Login

M2: Real Time Active Covid Cases

M3: Slot Booking M4: User Support

Test Case DesignObjective:

Testing Strategy:

Testing Criteria:

Dependencies:

Risks or Assumptions:

Manual TestingModule 1 - Main Screen, New user registration and login window popupsTest Case 1:

Test Case 2: Test Case 3:

Module 2 - Real time active Covid casesTest Case 1:

Module 3 - Slot RegistrationTest Case 1:

Module 4 - SupportTest Case 1:

Vaccination Management systemCovid-19

COVID-19 VACCINATION MONITORING SYSTEM

FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE TECHNOLOGY, Delhi-NCR

A PROJECT REPORT Submitted by ISHAN THAKUR [Reg No: RA1911003030068] AKHIL ABHILASH [Reg No: RA1911003030086] HARSHIT TYAGI [Reg No: RA1911003030093] Under the guidance of Mrs. Bharti Vidhury (Professor, Department of Computer Sciene & Engineering) in partial fulfillment for the award of the degree of **BACHELOR OF TECHNOLOGY COMPUTER SCIENCE ENGINEERING**

SRM INSTITUTE OF SCIENCE TECHNOLOGY (Under Section 3 of UGC Act, 1956) **BONAFIDE CERTIFICATE** Certified that this project report titled "COVID-19 VACCINATION MONITORING SYSTEM" is the bonafide work of AKHIL ABHI-LASH, HARSHIT TYAGI, ISHAN THAKUR, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

NOV 2022

Mrs. Bharti Vidhury

GUIDE

Professor

Dept. of Computer Sciene & Engi-

Signature of the Internal Examiner

SIGNATURE

Dr. R.P.Mahapatra

HEAD OF THE DEPARTMENT

Dept. of Computer Science Engineer-

Signature of the External Examiner

ABSTRACT

CVMS is designed to simplify the COVID - 19 Vaccination process that's running throughout the world and help to boost the efficiency and focuses on developing a software which will help people to keep track of their COVID-19 Vaccination Date, Time Slot, Place and will also remind them to complete the process. Software will also keep count on the total number and type of vaccines distributed. CVMS system will ensure that the vaccination process will be done in a systematic and efficient manner. CVMS development team is committed to privacy and being transparent about government requests for customer data globally. The CVMS system is developed using the Iterative model. Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my guide, Mrs. Bharti Vidhury her valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of her busy schedule, she has extended cheerful and cordial support to us for completing this project work.

Author

iv

TABLE OF CONTENTS
ABSTRACT iii
ACKNOWLEDGEMENTS iv
ABBREVIATIONS viii
1 INTRODUCTION ix
1.1 Outline: ix
1.2 Strategy:ix
2 Stakeholder And Process Model Overview x
2.1 Stakeholder:x
2.2 ProcessModel: xi
3 Identifying The Requirements xii
3.1 System Requirement: xii
3.2 Software Requirement: xii
3.3 Functional Requirement: xiii
3.4 Non Functional Requirement: xi
4 Work Breakdown Structure xv
4.1 Approach: xv
5 Architecture Design xvi
5.1 System Architecture: xvi
5.2 UseCase: xvi
5.3 EntityRelation: xvii
5.4 Dividing into Classes: xvii
5.5 DataFlow: xviii
5.6 Communication Diagram: xix
N.

6 Behavioral Pattern xx
6.1 SoftwareState: xx
6.2 EventScenarios: xxi
6.3 Deployment Diagramming: xxi
6.4 FrontEnd:xxii
7 Module Implementation xxiii
7.1 M1: Main Screen and New User Registration/Login xxiii
7.2 M2: Real Time Active Covid Cases xxviii
7.3 M3:SlotBookingxxxi
7.4 M4:UserSupportxxxiii
8 Test Case Design xxxiv
8.1 Objective: xxxiv
8.2 TestingStrategy: xxxiv
8.3 TestingCriteria: xxxiv
8.4 Dependencies:xxxv
8.5 Risks or Assumptions:
9 Manual Testing xxxvi
9.1 Module 1 – Main Screen, New user registration and login window pop-
upsxxxvi
9.1.1 TestCase1:xxxvi
9.1.2 TestCase2:xxxvii
9.1.3 TestCase3:xxxvii
9.2 Module 2 – Real time active Covid cases
9.2.1 TestCase1:xxxix
9.3 Module 3 – Slot Registration xl
9.3.1 TestCase1:xl
9.4 Module4–Support xli
9.4.1 TestCase1:xli
A Vaccination Management system xlii

A.1 Covid-19 xlii

ABBREVIATIONS CVMS COVID-19 Vaccination Monitoring System **UX User Experience** IDE Integrated Development Environment

WBS Work Breakdown Structure

MS Microsoft Office **OOP Object Oriented Programming ER Entity Relationship** DFD Data Flow Diagram AUIDN Aadhar Unique Identification Numbers UML Unified Modelling Language

BPMN Business Process Modelling Notation

SDLC Software Development Life Cycle GUI Graphical User Interface HTTP Hyper Text Transfer Protocol HTML Hyper Text Markup Language XML Extensible Markup Language **INR Indian Rupee QR Quick Response** viii

CHAPTER 1 INTRODUCTION 1.1 Outline:

CVMS project is exclusively designed for the government and respected citizens to simplify the COVID - 19 Vaccination process that's running throughout the world. We are so busy in our daily lives right now that it becomes impossible to keep track of things. To address this issue, this project will help the government to automate the patient selection process and will boost the efficiency and focuses on developing a software which will help people to keep track of their COVID-19 Vaccination Date, Time Slot, Place and will also remind them to complete the process. Software will also keep count on the total number and type of vaccines distributed.

1.2 Strategy:

Our strategy is to project best in industry customer service, and the current situation does not reflect this. The new CVMS system will ensure that the vaccination process will be done in a systematic and efficient manner. CVMS development team is committed to privacy and being transparent about government requests for customer data globally.

CHAPTER 2
STAKEHOLDER AND PROCESS MODEL OVERVIEW
2.1 Stakeholder:

2.2 Process Model:

The CVMS system will be processed using the Iterative model. Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

Χİ

xiii

CHAPTER 3
IDENTIFYING THE REQUIREMENTS
3.1 System Requirement:
3.2 Software Requirement:

3.3 Functional Requirement:

3.4 Non Functional Requirement: xiv

CHAPTER 4

WORK BREAKDOWN STRUCTURE

4.1 Approach:

A Work Breakdown Structure (WBS) is a deliverable-oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables.

The project is being divided into small modules until it is not further divisible. The root of the tree of work breakdown structure is labelled by the project name. For constructing a work breakdown structure, each node is recursively divided into smaller sub-modules, until the leaf level. the modules become undividable and independent. It follows a Top-Down approach.

CHAPTER 5

ARCHITECTURE DESIGN

5.1 System Architecture:

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

5.2 Use Case:

Use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

5.3 Entity Relation:

Illustrates how "entities" such as people, objects or concepts relate to each other within a system.

5.4 Dividing into Classes:

Classe are the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has an appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in xvii

their context. It describes various kinds of objects and the static relationship between them.

5.5 Data Flow:

To analyse the flow of data through a system or a process. It also gives insight into the inputs and outputs of each entity and the process itself. xviii

5.6 Communication Diagram:

Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. Used to portray the object's architecture in the system.

xix

CHAPTER 6
BEHAVIORAL PATTERN
6.1 Software State:

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioural diagram and it represents the behaviour using finite state transitions.

6.2 Event Scenarios:
Depicts the interaction between objects in a sequential order i.e., the order in which
these interactions take place. 6.3 Deployment Diagramming:
In the context of the Unified Modelling Language (UML), a deployment diagram falls
under the structural diagramming family because it describes an aspect of the system
itself. It describes the physical deployment of information generated by the software
program on hardware components. xxi
· · · · · · · · · · · · · · · · · · ·
6.4 Front End:
A Front-End diagram is a type of interaction diagram because it describes how a group
of objects works together. xxii
CHAPTER 7
MODULE IMPLEMENTATION CVMS Project is coded in Python Language so Installing Tkinter Module provides a
fast and easy way to create GUI applications
7.1 M1: Main Screen and New User Registration/Login
and a
xxiv
XXV
xxvi

7.2 M2: Real Time Active Covid Cases xxviii

xxvii

7.3 M3: Slot Booking

xxxi

xxxii

7.4 M4: User Support

XXXIII

CHAPTER 8

TEST CASE DESIGN

8.1 Objective:

To test the interface, front end and integration of all functions to find the smallest bug to rectify it to make the software error free to provide Cent efficient software for Covid Vaccine Monitoring to the Government and Public.

8.2 Testing Strategy:

Testing will be done by dividing software into small modules. And each module will be executed with all possible test cases.

8.3 Testing Criteria:

8.4 Dependencies:

Module 1: Depends on the user provided data.

Module 2: Depends on the statistics provided by various institutions.

Module 3: Depends on the slot availability and hospital limits.

Module 4: Depends on the working conditions of external sources.

8.5 Risks or Assumptions:

Slot unavailability can affect the working of module 3. Slow internet connections can Delay the opening of support websites which will affect module 4.

XXXV

CHAPTER 9

MANUAL TESTING

9.1 Module 1 – Main Screen, New user registration and

login window popups

9.1.1 Test Case 1:

Covid Vaccination Monitoring System main screen layout and interface.

9.1.2 Test Case 2:
User registration window + Detail entry + Registration function check
9.1.3 Test Case 3:
User Login window
xxxvii

Test Case 3.1: Successful login Test Case 3.2: Successful login xxxviii

Test Case 3.3:
User not found
9.2 Module 2 – Real time active Covid cases
9.2.1 Test Case 1:
Real time active SARS COVID cases window
xxxix

9.3 Module 3 – Slot Registration9.3.1 Test Case 1:Slot registration window + Data fill up + confirm slot registration xl

9.4 Module 4 – Support9.4.1 Test Case 1:Check the hyperlinks and confirm the working conditions for the same.xli

APPENDIX A VACCINATION MANAGEMENT SYSTEM A.1 Covid-19

Everyone, everywhere, should have access to COVID-19 vaccines. WHO is determined to maintain the momentum for increasing access to COVID-19 vaccines and will continue to support countries in accelerating vaccine delivery, to save lives and prevent people from becoming seriously ill. Countries should continue to work towards vaccinating at least 70 percent of their populations, prioritizing the vaccination of 100 percent of health workers and 100 percent of the most vulnerable groups, including people who are over 60 years of age and those who are immunocompromised or have underlying health conditions.

REFERENCES

- 1. Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin Series)
- 2. Design Patterns: Elements of Reusable Object-Oriented Software (Addison-Wesley Professional Computing Series) (Old Edition)
- 3. https://youtu.be/uJpQlyTCK4 xliii