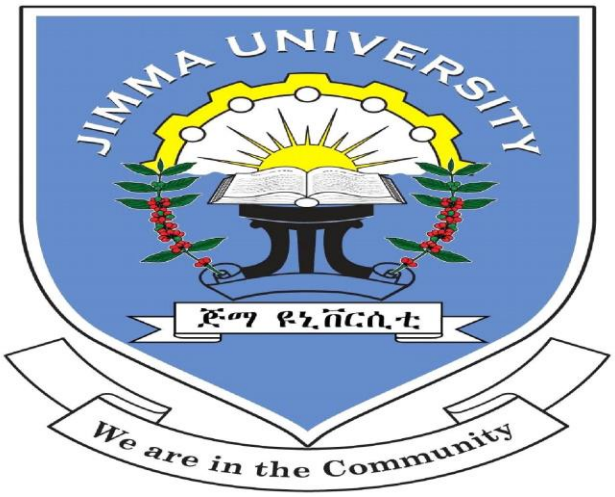


JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SOFTWARE ENGINEERING PROGRAM

CBTP PHASE-IV

HIRMATA MANTINA KEBELE DATA MANAGEMENT SYSTEM

May, 2023

Jimma, Oromia, Ethiopia

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SOFTWARE ENGINEERING PROGRAM

CBTP PHASE-IV

HIRMATA MANTINA KEBELE DATA MANAGEMENT SYSTEM

Declaration by Advisors

1. Mr. Nahil Kebede _____
2. Mrs. Tsedania D _____

Student List

No	Name	ID
1	Abdurahman Sheh Anwar	RU 0370/12
2	Abel Kinfu	RU 1372/12
3	Amansisa Gutu	RU 1254/12
4	Amir Abdulrzak	RU 5856/11
5	Atnabon Deressa	RU 5065/12
6	Bati Sileshi	RU 1426/12
7	Bernabas Tekalign	RU 1301/12
8	Betelhem Getachew	RU 1176/12
9	Bisrat Mengistu	RU 1829/12

Table of Contents

1. INTRODUCTION 1
1.1. Purpose of the document 1
1.2. Scope of the document 2
1.3. Project objectives 2
1.4. Overview of the project 3
2. DEPLOYMENT PLAN 4
2.1. Deployment Environment 4
2.2. Configuration Management 4
2.3. Backup and Recovery Plan 5
2.4. Rollback Plan 5
2.5. Deployment Process 5
3. TESTING PLAN 6
3.1. Testing Environment 6
3.2. GUI Checkpoint for multiple objects 8
Validation test cases 10
Authentication test cases 11
Database Checkpoint for default Check 12
Database checkpoint for custom check 13
4. MAINTENANCE PLAN 155
5. CONCLUSION 16

1. INTRODUCTION

1.1. PURPOSE OF THE DOCUMENT

The Hirmata Mantina Kebele Data Management System project is a critical initiative that aims to transform the way data is collected and managed in the kebele. The project seeks to develop a comprehensive data management system that can support the delivery of basic public services to citizens, while also collecting data on residents and providing them with essential identification documents such as Kebele ID, Kebele house registration, and birth certificates.

The implementation and testing plan for this project will be a vital document that outlines the strategies, timelines, resources, and procedures required for the successful deployment of the data management system. The plan will serve as a guide for all stakeholders involved in the project, including project managers, developers, testers, and end-users, ensuring that they are aligned and working towards a common goal.

One of the critical components of the implementation and testing plan is the integration of various modules of the data management system. These modules include the healthcare module, education module, water supply and sanitation module, and the identification and registration module. Each of these modules is responsible for delivering specific services and functions, and they must be integrated seamlessly to ensure that the data management system operates effectively.

The implementation and testing plan will also outline the timeline for the deployment of the data management system. This will include the various stages of the project, such as design, development, testing, deployment, and maintenance. The timeline will help to ensure that the project is completed within the specified timeframe and that all stakeholders are aware of their roles and responsibilities.

In addition to the implementation plan, the testing plan will also be an essential component of this document. The testing plan will define the test scenarios, procedures, and expected results for each module of the data management system. It will outline the testing strategies, resources required, and timelines for each stage of testing. The testing plan will help to ensure that the system is thoroughly tested, and any defects or issues are identified and resolved before deployment.

Overall, the purpose of this document is to provide a comprehensive plan for the implementation and testing of the Hirmata Mantina Kebele Data Management System project. The document will help to ensure that the project is completed successfully and that the data management system can support the sustainable development of the kebele and improve the lives of its residents.

1.2. SCOPE OF THE DOCUMENT

This document outlines the implementation and testing plan for the Hirmata Mantina Kebele Data Management System project. The project aims to develop a comprehensive data management system for the Hirmata Mantina Kebele, which is located in the Jimma zone of the Oromia region in Ethiopia. The system will support the delivery of basic public services to citizens and also collect data on residents and provide them with essential identification documents such as Kebele ID, Kebele house registration, and birth certificates.

This document outlines the scope of the project and the various components that will be included in the system.

1.3. PROJECT OBJECTIVES

The primary objective of the project is to develop a data management system that can efficiently collect and manage information on the residents of Hirmata Mantina Kebele. The system will enable kebele officials to provide basic public services, manage citizen information, and issue essential identification documents such as Kebele ID, Kebele house registration, and birth certificates. The system will also help identify key issues and trends, track progress towards development goals, and support evidence-based decision-making.

The scope of the system includes the following components:

1. Kebele ID, Kebele House Registration, and Birth Certificates: The system will issue essential identification documents to residents of the kebele. This will help kebele officials to maintain accurate records of residents and enable citizens to access various services and benefits.

2. Document Management: The system will allow officials to upload and manage documents related to kebele administration, such as meeting minutes, policies, and guidelines. Documents can be categorized, tagged, and easily accessed by authorized users.

3. User Training and Support: The project team will provide training to the users on how to use the new system. The team will also provide ongoing support to users and address any issues or concerns that arise.

4. System Development and Integration: The development phase will involve the coding and testing of the system components. The team will integrate the different modules of the system and ensure that they function seamlessly.

5. Data Migration and Integration: The project team will ensure the migration of data from the existing systems to the new data management system. The team will also integrate the new system with other relevant systems and platforms.

6. The testing plan for the system, including test scenarios, procedures, and expected results

7. The risks and challenges associated with the implementation of the project and how they will be addressed

The project will involve the collaboration of various stakeholders, including kebele officials, citizens, and IT professionals. The project team will work closely with these stakeholders to ensure that the system meets their needs and expectations.

The project will have a significant impact on the kebele by improving the delivery of basic public services and enhancing the collection and management of resident data. The project will also provide citizens with essential identification documents, which will enable them to access various government services and programs.

In conclusion, the Hirmata Mantina Kebele Data Management System project is a critical initiative that will transform the way data is collected and managed in the kebele. The project will have a significant impact on the lives of citizens and residents, and will contribute to the sustainable development of the kebele.

1.4. OVERVIEW OF THE PROJECT

The Hirmata Mentina Kebele Management System is a web-based application implemented using the Django framework. The project seeks to develop a comprehensive data management system that can support the delivery of basic public services to citizens, while also collecting data on residents and providing them with essential identification documents such as Kebele ID, Kebele house registration, and birth certificates.

The project's main objective is to digitalize and centralize key information, facilitate communication and collaboration among kebele officials, and provide an efficient and user-friendly platform for managing and organizing the administrative activities of the kebele. Additionally, the system aims to improve the delivery of basic public services to citizens by collecting data on residents and providing them with essential identification documents such as Kebele ID, Kebele house registration, and birth certificates.

Features

1. User Authentication and Access Control: Secure user authentication with specific access permissions for administrators, kebele officials, and citizens.

2. Citizen Registration and Management: Citizens can register in the system, provide personal information, and their contact details.

3. Services and Requests: Citizens can submit requests for various services offered by the kebele, and officials can process these requests, track their status, and update citizens on progress.

4. Document Management: Officials can upload and manage documents related to kebele administration, such as meeting minutes, policies, and guidelines.

5. Communication and Notifications: The system facilitates communication among kebele officials, citizens, and other stakeholders via email or SMS notifications.

6. Dashboard and Administrative Tools: A comprehensive dashboard for administrators, providing an overview of key metrics, system health, and user activities, along with administrative tools for user role management, system configurations, and data backups.

Technology Stack

The Hirmata Mentina Kebele Management System is built on the Django web framework, which utilizes Python as the primary programming language. The system may use a combination of front-end technologies such as HTML, CSS, and JavaScript for the user interface. It may employ a relational database management system (RDBMS) PostgreSQL to store and retrieve data efficiently.

2. DEPLOYMENT PLAN

2.1. Deployment Environment

The deployment environment for the Hirmata Mantina Kebele Data Management System project consists of a server that will host the Django application and the PostgreSQL database. The server will be running on a Linux operating system, specifically Ubuntu 20.04 LTS. The server specifications include a 2.0 GHz processor, 8 GB of RAM, and 500 GB of storage space.

2.2. Configuration Management

Configuration management for the Hirmata Mantina Kebele Data Management System project will be handled using Ansible. Ansible is an open-source automation tool that can be used to automate software provisioning, configuration management, and application deployment.

Ansible will be used to configure the server, install dependencies, and set up the PostgreSQL database. Additionally, Ansible will be used to manage the Django application configuration files.

2.3. Backup and Recovery Plan

A backup and recovery plan is crucial for ensuring that data is not lost in case of system failure or data corruption. The backup plan for the Hirmata Mantina Kebele Data Management System project will involve creating regular backups of the PostgreSQL database. These backups will be stored on a separate server or on an external storage device. The recovery plan will involve restoring the backup to the server in case of data loss or corruption.

2.4. Rollback Plan

A rollback plan is important for ensuring that the application can be rolled back to a previous version in case of errors or bugs in the new version. The rollback plan for the Hirmata Mantina Kebele Data Management System project will involve creating a backup of the current version of the application and the PostgreSQL database before deploying the new version. If any issues arise, the backup will be restored and the previous version of the application will be redeployed.

2.5. Deployment Process

The deployment process for the Hirmata Mantina Kebele Data Management System project will involve several steps to ensure smooth and efficient deployment.

1.Code Push: The first step in the deployment process is to push the code to a version control repository such as GitHub. This ensures that all the code changes are tracked, and multiple team members can work on the project simultaneously.

2. Continuous Integration: Once the code is pushed to the repository, a continuous integration (CI) process will be initiated. The CI process involves building the code, running unit tests and integration tests, and generating various reports such as test coverage and code quality reports. This step ensures that the code is error-free, meets the quality standards, and can be deployed without any issues.

3 Continuous Deployment: After the code passes the CI process, a continuous deployment (CD) process will be initiated using a tool such as Jenkins or GitLab CI/CD. The CD process will automate the deployment of the application to the production environment. The CD tool will be configured to deploy the code to the server, update the application database schema, and restart the application server.

4 Testing: Once the application is deployed, extensive testing will be performed to ensure that the application works as expected. The testing process includes functional testing, load testing, and security testing. This step ensures that the application is stable and can handle various types of traffic and user request.

5 Monitoring and Maintenance: After the application is deployed and tested, it will be continuously monitored to ensure that it runs smoothly and efficiently. Any issues or errors will be resolved promptly, and the application will be maintained to ensure that it meets the user requirements and quality standards.

By following this deployment process, the Hirmata Mantina Kebele Data Management System project can ensure smooth and efficient deployment of the application, reduce errors and bugs, and improve the overall application quality.

3. TESTING PLAN

The Testing Plan for the Hirmata Mantina Kebele Data Management System project will ensure the quality of the software before it is deployed to production. Testing will be performed at different stages of the software development life cycle to ensure that the system is functioning as intended.

3.1. Testing Environment

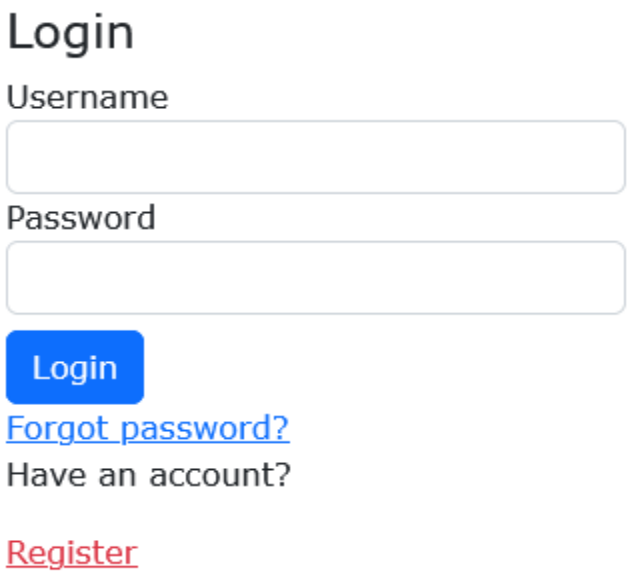
The testing environment for this project will consist of the development environment and the staging environment. The development environment will be used for testing individual components of the system, while the staging environment will be used for end-to-end testing.

GUI checkpoint for a single property

This type of test case would focus on verifying that all the necessary information related to a single resident data, such as the resident info, resident address, family, ID card and other relevant information, is displayed correctly on the website's GUI.

Test Case 1: Verification of Login Button Label

i. Open the Hirmata Mentina KDMS login page.
ii. Verify that the "Login" button label is displayed correctly.
iii. Ensure that the label is aligned with the button and has the correct font and font size.



Test Case 2:Verification of Get Started Button

i. Open the Hirmata Mentina KDMS welcome page.

ii. Verify that the "Get Started" button label is displayed correctly.

iii. Ensure that the label is aligned with the button and has the correct font and font size.

GUI Checkpoint for a single object/window

This type of test case focuses on verifying that a specific object or window, such as a button, menu item, text box, or dialog box, is displayed correctly and functions properly, with no layout or alignment issues, missing or incorrect text, or other defects.

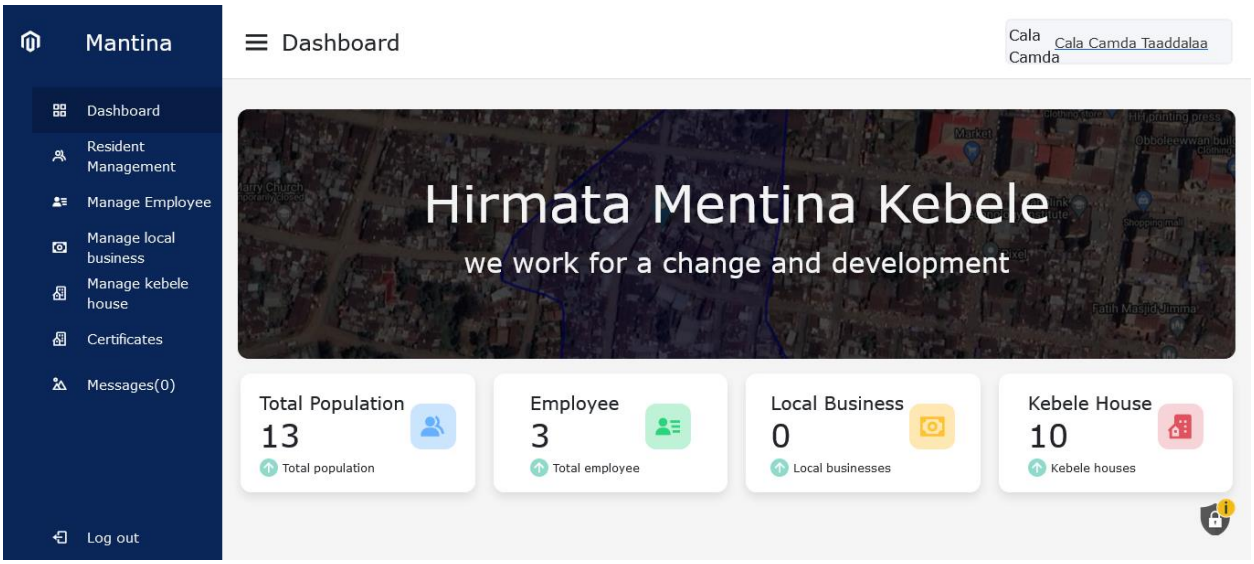
Test Case 1: Verify Object/Window Visibility

Open the Hirmata Mentina KDMS.

Navigate to a specific object or window.

Verify that the object or window is visible to the user.

Ensure that the object or window does not obstruct other objects or windows.



3.2. GUI Checkpoint for multiple objects

This type of test case focuses on verifying that multiple objects or windows on the website, such as a navigation menu or a pop-up window, are displayed correctly and function properly when the user interacts with them.

Test Case 1: Verification of Search Results Page

i. Enter a search query in the search bar.

ii. Verify that the search results page is displayed correctly.

iii. Ensure that the search results are displayed in the correct order and have the correct font and font size. Verify that the user bar and other elements on the page are aligned correctly.

iv. Pagination works correctly.

Functional test cases

Functional test cases are designed to test the system's functional requirements to ensure that it performs as intended. These test cases evaluate the system's functions or features, such as input validation, calculation accuracy, and output formatting. The purpose of functional testing is to ensure that the system meets its functional requirements and provides a satisfactory user experience.

Validation test cases

Test case 1: Validating name entry

We used regular expression validation mechanism so that a person can only enter characters in name entry. Entering numbers or other kind of symbols is not allowed.

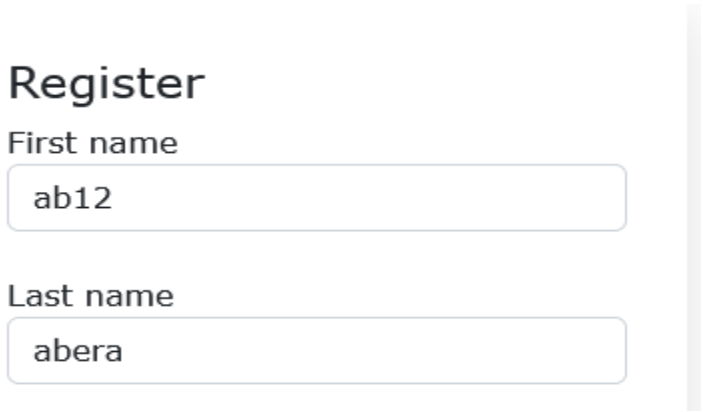
Functional test cases

Functional test cases are designed to test the system's functional requirements to ensure that it performs as intended. These test cases evaluate the system's functions or features, such as input validation, calculation accuracy, and output formatting. The purpose of functional testing is to ensure that the system meets its functional requirements and provides a satisfactory user experience.

Validation test cases

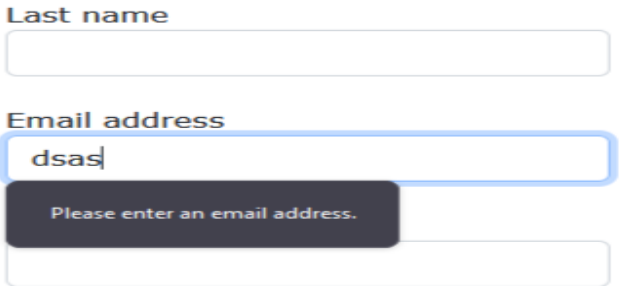
Test case 1: Validating name entry

Entering numbers or other kind of symbols is allowed. Which is not correct so this fails the test case.



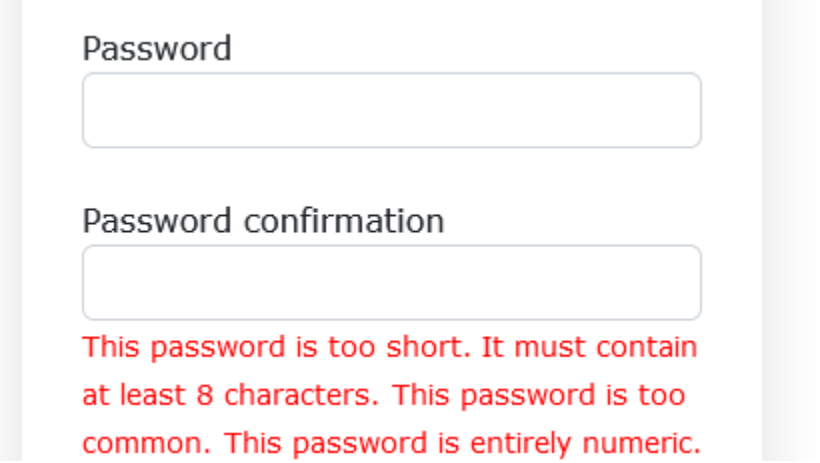
Test case 2: Validating email entry

Browser have their own validation for email imputs to validate email. If the entered data has a valid email format(having string @, .), it will be valid. Else, it will be invalid and fails.



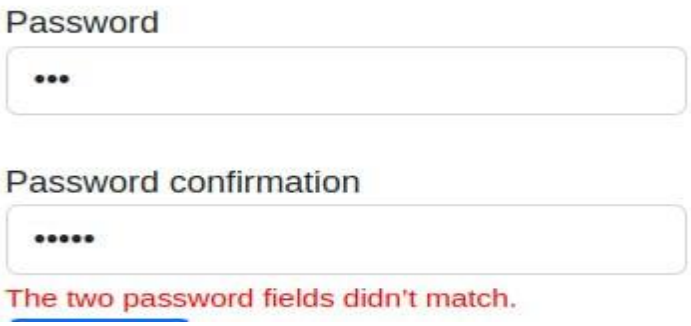
Test case 4: Validating password entry

We took this test case seriously so that the strength of the password becomes much stronger. The password must be strong to be valid. To be valid, It must minimum length of 8 characters. Otherwise, it will be weak and fails.



Test case 5: Validating password match

The password entered in password and confirm-password input filed must be the same. Otherwise, the user can't proceed and has to re-enter password that matches.

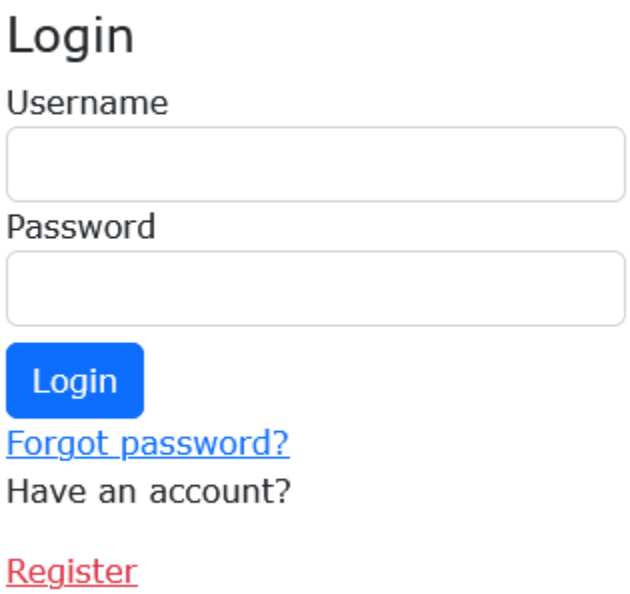


Authentication test cases

It is one of the most important tests that must be carried out seriously. These test cases are very important for the integrity, privacy and security. The system must ensure that a person who is claiming to login, is the right user.

Test case 1: Authentication

If the user has account and his credentials are correct, he will successfully login. Else, he will not be authenticated to enter into the system.



Test case 2: Password hashing

We used crypt hashing algorithm for encrypting the user's passwords. Whenever a user is added to the system, his password will be stored on the database in ha hashed for so that nobody can understand and decrypt it. So it passed the password hashing test.

Database Checkpoint for default Check

It is a type of functional testing that verifies the correctness of the default values stored in a database. It ensures that the database has been initialized correctly and contains all the expected data with the expected values in each field.

The purpose of this type of testing is to ensure that the database is consistent and conforms to the defined standards. It helps identify any issues or errors that may arise due to incorrect initialization or missing data. By performing this type of testing, it is possible to ensure that the website operates as expected, and the data is accurate and reliable.

Test Case 1: Database Checkpoint for default date check

Our database schema is configured with the help of timestamps so that whenever new data is inserted, it records a date and time of the request by default. So by default, it stores default date(now), and updates date when there is a change.

Database checkpoint for custom check

It is a type of functional testing that verifies the correctness of the data stored in a database against specific user-defined criteria. It involves checking whether the data in the database meets the expected values, format, and structure based on the user's requirements.

Test case 1. Verify that the data of resident in the database matches the name provided in the resident database.

i. Create a new resident data form for a resident.
ii. Submit the form and wait for the data to be stored in the database.
iii. Access the database and query the resident’s data.
iv. Compare the value in the database to the value provided in the resident data form.
v. Pass the test case if the values match, fail the test case otherwise.

Test Case 2. Verify Custom Search Results

1. Open the Hirmata Mentina KDMS .
2. Navigate to the screen for searching for residents data.
3. Perform a search with id.
4. Verify that the custom search results are fetched and displayed correctly.

Database checkpoint for runtime record check

Test Case 1: Verify that the information displayed on the website matches the information stored in the database.

Test Cases Execution

The test cases will be executed manually and automatically. The manual testing will be done by the Quality Assurance (QA) team to ensure that the software is functioning as expected. The automated testing will be done by the continuous integration and continuous deployment (CI/CD) pipeline to ensure that the software is being built, tested, and deployed correctly.

Test Data and Scripts

Test data will be generated to test the functionality of the system. This test data will be used to ensure that the system is working as expected and that it meets the requirements of the client. Test scripts will also be generated to automate the testing process.

Test Results

The test results will be recorded and analyzed to identify any defects or issues in the software. The results will be used to improve the software and ensure that it is functioning as intended. Any defects or issues identified during testing will be fixed and retested until the software meets the requirements of the client.

4. MAINTENANCE PLAN

A maintenance plan is a set of procedures and policies that define how a system or application will be maintained. The purpose of a maintenance plan is to ensure that the system or application remains functional and effective over time. The maintenance plan for the Hirmata Mentina Kebele Management System outlines the maintenance environment, process, change management, bug tracking and resolution, and performance monitoring and tuning.

Maintenance Environment:

The maintenance environment refers to the infrastructure, tools, and resources required to maintain the Hirmata Mentina Kebele Management System. This includes hardware, software, and network components, as well as the development, testing, and production environments. The maintenance environment must be carefully designed and managed to ensure that maintenance tasks can be performed efficiently and effectively.

Maintenance Process:

The maintenance process defines the steps and procedures required to maintain the Hirmata Mentina Kebele Management System. This includes regular maintenance tasks, such as backups and updates, as well as more complex tasks, such as bug fixes and feature enhancements. The maintenance process must be well-documented and regularly reviewed to ensure that it remains effective over time.

Change Management:

Change management refers to the process of managing changes to the Hirmata Mentina Kebele Management System. This includes changes to software, hardware, network components, and other system elements. The change management process must be carefully designed to ensure that changes are thoroughly tested and reviewed before they are implemented. This helps to reduce the risk of system downtime and data loss.

Bug Tracking and Resolution: Bug tracking and resolution refers to the process of identifying and fixing bugs or errors in the Hirmata Mentina Kebele Management System. Bugs can cause system crashes, data loss, and other issues that can impact system performance and reliability. The bug tracking and resolution process must be well-defined and regularly reviewed to ensure that bugs are identified and fixed in a timely and efficient manner.

Performance Monitoring and Tuning: Performance monitoring and tuning refers to the process of monitoring the performance of the Hirmata Mentina Kebele Management System and adjusting improve performance. This includes monitoring system resource usage, network traffic, and application response times, among other factors. The performance monitoring and tuning process must be carefully designed to ensure that the system remains responsive and reliable over time.

5. CONCLUSION

The Hirmata Mentina Kebele Management System is a comprehensive web-based application developed using the Django framework. It aims to streamline and automate various administrative tasks and processes within the context of a kebele (a local administrative unit) in the fictional town of Hirmata Mentina. The system provides a user-friendly platform for managing and organizing the administrative activities of the kebele, digitizing and centralizing key information, facilitating communication and collaboration among kebele officials, and enhancing decision-making processes.

Summary of the deployment and testing plan:

The deployment of the Hirmata Mentina Kebele Management System will require a comprehensive testing plan to ensure the system is fully functional and meets the needs of the kebele. The testing plan will include unit testing, integration testing, and user acceptance testing. Unit testing will involve testing individual components of the system to ensure they function correctly. Integration testing will involve testing the integration of multiple components to ensure they work together as intended. User acceptance testing will involve testing the system with end-users to ensure it meets their needs and is user-friendly.

The deployment of the system will require careful consideration of the environment in which it will be deployed. The system will need to be deployed on servers that meet the minimum system requirements and are configured to optimize performance. The deployment environment will need to be secure, with appropriate measures in place to protect against cyber threats and data breaches.

Recommendations for future improvements:

The Hirmata Mentina Kebele Management System has been designed to meet the needs of the kebele, but there is always room for improvement. Some potential areas for future improvement include:

1. Mobile compatibility: The system could be made more accessible by developing a mobile application that would allow citizens to access the system from their smartphones.

2. Artificial intelligence (AI) integration: The system could be enhanced by integrating AI technologies, such as chatbots or machine learning algorithms, to improve the efficiency of certain processes.

3. Improved reporting: The system could be enhanced by developing more detailed and customizable reporting features, allowing officials to generate reports on specific metrics and areas of interest.

4. Enhanced data analytics: The system could be enhanced by developing more advanced data analytics features, allowing officials to generate insights from the data collected by the system.

5. Multilingual support: The system could be made more accessible by developing multilingual support, allowing citizens to access the system in their preferred language.

6. Integration with other systems: The system could be enhanced by integrating with other systems, such as financial management or procurement systems, to improve the overall efficiency of kebele operations.

In conclusion, the Hirmata Mentina Kebele Management System is a comprehensive web-based application that has been designed to streamline and automate various administrative tasks and processes within the context of a kebele. The deployment and testing plan for the system will need to be carefully considered to ensure it is fully functional and meets the needs of the kebele. There is also room for future improvement, and potential areas for future improvement include mobile compatibility, AI integration, improved reporting, enhanced data analytics, multilingual support, and integration with other systems.