Addis Ababa Institute of Technology

School of Information Technology and Engineering

Department of IT/SW Eng.

Efoyta Doctor's Appointment Booking System

Test Plan Document

Section 2

Group 4 Team Members:

1. Biniyam Assefa Mekonnen····· UGR/8320/14
2. FikreYohannes Abera Shibru····· UGR/8889/14
3. Isam Ahmed Hussen····· UGR/7643/14
4. Kaleb Asratemedhin Bekele····· UGR/9104/14
5. Meklit Asrat Tefera····· UGR/5387/14
6. Daniel Demerew Endale····· UGR/9530/14

Advisor: Ms. Nuniyat Kifle

Date: January 31, 20

List of Figures

Introduction

The purpose of this testing document is to outline the testing approach for a Doctor's appointment system that allows patients to book appointments and pay bills through the system. The system aims to provide a convenient and streamlined process for patients to manage their healthcare needs. The testing process will encompass unit testing, integration testing, and system testing to ensure the reliability, functionality, and security of the system

1.1 Purpose of the document

The purpose of this testing document is to provide a comprehensive overview of the testing approach for the Doctor's appointment system. It serves as a reference guide for the testing team, outlining the different types of testing to be conducted and their importance in ensuring the reliability and functionality of the system. This document also helps to establish a clear understanding of the testing scope, objectives, and deliverables. By documenting the testing strategy, it enables effective communication and collaboration between the development team, testing team, and other stakeholders involved in the project. The document serves as a valuable resource for planning, executing, and tracking the progress of the testing activities, ensuring that all necessary tests are performed and any identified issues are addressed and resolved in a timely manner. Ultimately, the purpose of this document is to support the successful implementation of the Doctor's appointment system by ensuring that it meets the highest quality standards and provides a seamless experience for patients.

2. Features to be tested/not to be tested

In the testing plan, the focus on features to be tested revolves around the user's perspective, emphasizing functionality and user-oriented processes. Functional testing includes crucial tasks such as creating and updating user accounts, with the associated risks assessed on a simple scale (H, M, L). This user-centric approach ensures that the testing criteria align with the user's understanding of system functions, prioritizing tasks essential to their roles. On the other hand, nonfunctional requirements, spanning user interface, hardware, software, security, and performance, are also outlined for testing. These aspects, while not delving into technical details, emphasize the user experience, system reliability, and security measures. Conversely, the features not to be tested are identified with clear reasons, encompassing considerations like exclusion from the current software release, low risk due to stability in previous use, or features scheduled for release without testing in this specific version. This dual approach, considering both what to test and what not to test, provides a comprehensive view that aligns with both user expectations and version control needs.

2.1. Features to be tested

2.1.1 Functional Testing:

1. Create Account (FR01) :

Purpose: Validate the creation of a new user account, ensuring accurate data entry and successful account creation.

Verify the system allows the user to register with a unique username and password.

Confirm the system displays a message confirming the registration.

Test error handling by attempting to register an already existing user.

Risk Level: Medium

2. User Login(FR02):

Purpose: Validate the system's ability to allow existing users to log in securely.

Ensure the system authorizes access for users providing valid usernames and passwords.

Verify the system displays an error and notifies the user to try again for invalid login attempts.

Check the dependency on successful account creation (FR01).

Risk Level: Medium

3. Apply For Privileged Account (FR03)

Purpose**:** Enable Doctor’s and Lab technicians access privileged features of the system.

Test Cases: Confirm the system processes the request effectively with valid inputs.

Risk Level: medium

4. Search Doctor(FR05):

Purpose: Enable registered or non registered users to search for Doctors.

Test Cases: Confirm the system processes searches effectively with valid inputs.

Ensure the system displays the photo and the descriptions of the doctor searched.

Risk : Low

5. pay bill(FR 09):

Purpose: enable patients to pay bill online

Test case: confirm that the system accurately processes the requests for a pay bill with valid inputs.

Ensures that the system property restricts access for unauthorized user attempting to utilize the pay bill system and the use patient credentials.

Risk Level : High

6. Request for Lab test (FR08):

Purpose: enable Doctor asses a lab test

Test case: Confirm the system processes searches effectively with valid inputs.

Risk Level: Medium

7.Send lab result(FR11):

Purpose: Enable Lab technician send the result to Patient and Doctor

Test Cases: Confirm the system processes the request effectively with valid inputs like patient Id and Doctor Id.

Risk Level: Medium

2.1.2 Nonfunctional Requirements Testing:

- **User Interface:**Assess the ease of use and user-friendliness of the interface, with a particular emphasis on the arrangement of elements, ease of navigation, and visual design.

  Risk Level : High

- **Hardware and Software Compatibility:** Verify the seamless performance and compatibility of the system across various devices and platforms.

  Risk Level: Medium

- **Security:** Verify the proper implementation of security measures, including the authentication process for users..

  Risk Level: High

- **Performance:** Validate the system's performance efficiency across different scenarios, paying particular attention to response time, load capacity, and resource utilization..

Risk Level : High

2.2. Features not to be tested

- Not to be included in this release of the Software:
    - Certain features that are planned for future releases or enhancements may not be tested in the current release due to prioritization and development timelines.
- Low risk, has been used before and is considered stable:
    - Features that have been thoroughly tested in previous releases and have a low risk of introducing defects may not undergo extensive testing to optimize resource allocation.
- Will be released but not tested or documented as a functional part of the release of this version of the software:
    - Some features might be included in the release but are intentionally excluded from testing due to various reasons, such as time constraints or a deliberate decision to release them in a beta phase for user feedback without guaranteeing full functionality.

3. Pass/Fail criteria

The success/fail criteria for testing the doctor appointment booking system are as follows:

- Pass Criteria:
    - If the actual output of a tested feature matches the expected result, the pass criteria are marked as "success" or "pass."

        - For instance, if a user successfully creates a new account, and the system responds as expected, the test is considered a pass.
- Fail Criteria:
    - If the actual output deviates from the expected result, indicating an inconsistency or error in the system, the pass/fail criteria is marked as "fail."
    - For example, if a user encounters an error while updating their account information, and the system response is not in line with the expected behavior, the test is considered a fail.

Adhering to this straightforward approach ensures clarity in evaluating the system's performance against the anticipated outcomes, enabling the testing team to make informed decisions based on whether the system meets the specified criteria or falls short.

4. Approach/Strategy

Integration Testing Strategy:

Objective(reason to use this approach) : The primary goal of integration testing for Efotya Booking and appointment is to ensure good communication and collaboration among different modules and components within the system. This testing strategy aims to identify and rectify any issues that may arise when integrated units are combined.

Approach: Integration testing for Efotya Booking and appointment will follow a systematic and incremental approach. The testing process will involve combining individual components or modules and testing them as a group, gradually progressing towards testing the entire system's integrated functionality.

Module Integration Testing:

Description: Individual modules are combined and tested to ensure their interfaces and interactions function correctly.

Tools: **Playwright**

Metrics: we measure the success of each module's integration based on the number of test cases passed.

Component Integration Testing:

Description: Multiple modules are integrated to form components, and their interactions will be thoroughly tested.

Tools: Playwright

Metrics: we assess the successful integration of components based on test case outcomes.

End-to-End Integration Testing:

Description: The complete Efotya Booking and appointment system is tested to verify the end-to-end functionality and smooth integration of all components.

Tools: **Postman, ThunderClient**

Metrics: we evaluate the success of the entire system's integration in terms of passed test cases.

Special Requirements for Integration Testing:

-Environment Setup:

It needs to Ensure that Node.js is installed and configured for the testing environment.

It needs to Verify the availability and connection of mysql for database testing.

-Version Compatibility:

It needs to confirm that installed Node.js versions align with project requirements to avoid compatibility issues.

Metrics Collection:

Metrics: Integration success rates, data flow between components, and overall system integration.

Level of Collection: Metrics will be collected at the integration testing level.

Configuration Management:

Handling: the configuration management is handled using npm or yarn for project configurations.

Number of Configurations: our system testing is conducted on configurations involving Node.js, mysql, and various testing libraries.

Testing Metrics and Reporting:

We measured the success of integration testing based on the percentage of passed test cases.

We identify and document any integration issues, including communication failures or data inconsistencies.

Challenges and Mitigations:

Payment Processing Simulation:

Explanation: In Efotya Booking and appointment, handling real financial transactions involves complexities such as security concerns, legal requirements, and the need for a live payment gateway, which may not be feasible during the testing phase.

Action Taken: To overcome these complexities and constraints, the testing team opts for simulating the payment process. This involves creating a controlled environment that mimics the essential aspects of payment processing without involving actual financial transactions. The

focus is on ensuring that the payment-related features and functionalities work as intended without executing real transactions.

Data Input Validation Challenges:

Explanation: Ensuring the validity and security of user inputs is critical for the reliability of the Efotya Booking and appointment. Challenges may arise in handling diverse input formats, preventing malicious input, and maintaining data integrity to meet system requirements.

Action Taken: The testing team focuses on comprehensive input validation scenarios specific to the travel system. This includes testing various input formats related to user registrations, search queries, and booking details.

Error Handling and Messaging Challenges:

Explanation: Effective error handling and clear, user-friendly error messages are crucial for user understanding and system recovery in the Efotya Booking and appointment. Challenges may involve inconsistent error messages, unclear instructions, or inadequate error prevention.

Action Taken: we tests error scenarios within the Efotya Booking and appointment, including incorrect inputs during account creation, and search processes.

This integration testing approach aligns with the goals of Efotya Booking and appointment, ensuring a fully integrated system while addressing specific challenges and constraints at the integration testing level.

5. Test cases with specifications

A test case is a set of input data and expected results that exercises a component with the purpose of causing failures and detecting faults.

Table 1: Test Case Specification Create Account

Name = Create Account				
Purpose=To authorize users to the system				
Test Data = Name (invalid name, Valid name, empty) User name(invalid User name, valid user name) Email (Valid Email, invalid Email) Phone number(invalid User name, valid user name) Confirm Email (Valid Email, invalid Email) Password (invalid password, valid password) Confirm Password (invalid password, valid password)				
Input	Expected result		utput	il
Empty name and all elds are valid	correctly formatted	name and Any valid elds	ls are required”	

Name other fields are valid	correctly formatted	=875333 <2 characters >50 characters id data for other fields		
User Name and elds are valid	correctly formatted me”	User Name and id data for other fields	ls are required””	
User Name and all elds are valid	correct user	Name is occupied name>25 characters	User Name is exists ”	

Empty email and all lds are valid	correct formatted	email and Any valid other fields	ls are required””	
Invalid email and all lds are valid	correct formatted	email and Any valid other fields	ail is incorrect”	
Confirmation Email all other fields are	does not match”	Confirmation Email ny valid data for other	ls are required”	
Confirmation Email r fields are valid	does not match”	l = abc@yahoo.com nfirmation email= ahoo.com y valid data for other	email format”	
password And all lds are valid	strong Password ngth greater than	password y valid data for other	ls are required”	

password And all elds are valid	strong Password ngth greater than	word< 3 y valid data for other	strong Password ngth greater than	
confirmation rd and all other e valid	ord does not	confirmation password y valid data for other	ls are required”	
confirmation rd and all other e valid	ord does not	vious password != ation password ll the other fields are	ord does not	
s are empty	ls are required”	ds are empty	s are required”	
fields	Account created ully”	id data for all fields	nt created sfully”	

Table 2: Test case specification Login

Name: login account				
Purpose: enable users to login to their account				
Test Data=Email (Valid Email, invalid Email) Password (Valid password, invalid Password)				
Input	**Expected result**	**Data**	**Actual output**	**Pass/fail**
All fields are empty	“fill all fields”	All fields are empty	“fill all fields”	Fail
Empty password and valid email	“please enter the password”	Empty password and valid email	“please enter the password”	Fail
Empty Email and valid password	“Enter correct formatted email ”	Empty Email and all other fields are valid	“Enter correct formatted email ”	Fail
Valid email and incorrect password	“Incorrect Password”	“incorrect Password”	“Incorrect password”	fail

Invalid Email and valid password	“Enter correct formatted email ”	If email doesn’t include @ and all other fields are valid	“Enter correct formatted email ”	Fail
All fields are valid	Redirected to discover page	Any valid data for the all fields	Redirected to discover page	Pass

Table 3: Test case specification Apply for Privileged Account

Name: Apply for Privileged Account
Purpose: to become a privileged user
Test Data = Username(invalid username, valid username, empty) Password(invalid password, valid password, empty) Department(invalid department, valid department, empty) Speciality(invalid speciality, valid speciality, empty) Personal information(invalid, valid, empty) Staff ID (invalid, valid, empty)

Input	Expected result	Data	Actual output	Pass/fail
Empty username and all other informations and confirmation are valid	“cannot leave empty”	Empty username and all other informations are valid	“cannot leave empty”	Fail
Invalid username and all other informations and confirmation are valid	“User name does not exist”	Invalid username and all other informations and confirmation are valid	“User name does not exist.”	Fail
Valid username and all other informations and confirmation are valid	"Displays a privileged confirmation page.”	Any valid data for both username and all other informations and confirmation are valid	"Displays a privileged confirmation page.”	Pass
Miss match password and all other informations and confirmation are valid	“Password doesn’t match”	If the password is not matching and all other informations and confirmation are valid	“Password doesn’t match”	Fail

Invalid password and all other informations and confirmation are valid	“Enter a correct password”	If password contains invalid characters and all other informations and confirmation are valid	“Enter a correct password”	Fail
Empty password and all other informations and confirmation are valid	“cannot leave empty password ”	Empty password and all other informations and confirmation are valid	“cannot leave empty password ”	Fail
Valid password and all other informations and confirmation are valid	"Displays a privileged confirmation page.”	any valid data password all other informations and confirmation are valid	"Displays a privileged confirmation page.”	Pass
Invalid department and all other informations and confirmation are valid	“Enter a valid department”	When the department doesn’t exist and all other informations and confirmation are valid	“Enter a valid department.”	Fail

Empty department and all other informations and empty confirmation are valid	“Department can’t be empty”	When the department is left empty and all other informations and empty confirmation are valid	“Department can’t be empty”	Fail
Valid department and all other informations and confirmation are valid	“Displays a privileged confirmation page.”	When both the department and all other informations and confirmation are valid	"Displays a privileged confirmation page.”	Pass
Invalid personal information all other informations and confirmation are valid	“Personal information doesn’t match”	When the personal information doesn’t match with the data base information and all other informations and confirmation are valid	“Personal information doesn’t match”	Fail
Valid personal information and all other informations and confirmation are valid	“Displays a privileged confirmation page.”	When the personal information and all other informations and confirmation are valid	“Displays a privileged confirmation page.”	Pass

Empty personal information all other informations and empty confirmation are valid	“Personal information can not be empty”	When the personal information is left empty and all other informations and empty confirmation are valid	“Personal information can not be empty”	Fail

Table 4: Test case specification for Book Appointment

Name: Book Appointment
Purpose: To let patient book an appointment
Test Data **Date (empty)** **startTime(empty)** **endTime(empty)** **patientId (Valid Id, invalid Id , empty)** **doctorId (Valid Id, invalid Id , empty)** **appointmentId (Valid Id, invalid Id , empty)**

Input	Expected result	Data	Actual output	Pass/fail
One or more empty	“please fill this field””	All fields are empty	“please fill all field”	Fail
valid patientId or doctorId	patientId does not exist”	valid patientId and any Valid confirmation patientId	patientId does not exist”	Fail
valid AppointmentId	AppointmentId d sn’t match”	valid AppointmentId nd Any Valid onfirmation ppointmentId	AppointmentId oesn’t match”	Fail
valid fields	ou have successfully oked Appointment.”	valid fields	ou have successfully oked Appointment.”	Pass

Table5 : Test case specification for requesting lab test

Name: Request lab test
Purpose: enable doctors to request for lab test

Test Data= patient information (valid,invalid,empty) Doctor's name(valid,invalid,empty)				
Input	**Expected result**	**Data**	**Actual output**	**Pass/fail**
All fields are empty	"fill all fields"	All fields are empty	"fill all fields"	Fail
Empty name And all other fields are valid	"Enter Correct name"	Empty name And any valid data for other fields	"Enter Correct name"	Fail
invalid name and all other fields are valid	"Name does not match"	If name doesn't match and Any valid data for other fields	"Name does not match"	Fail

Invalid patient information and valid name and valid confirmation	“Enter valid patient information”	When doctor’s name is valid and patient information is invalid and valid confirmation	“Enter valid patient information”	Fail
Empty patient information and valid doctor’s name	“Patient information can not be empty”	When patient information is empty and doctor’s name is valid	“Patient information can not be empty”	Fail
Valid doctor’s name and valid patient information	“Lab request successful."	When both doctor’s name and patient information are valid and valid confirmation	“Lab request successful."	Pass

Table 6 : Test case specification for ordering checkup

Name: **Order Checkup**
Purpose: To let doctors order a **Checkup**

Test Data

patientId (valid, invalid , empty)

doctorId (valid, invalid , empty)

checkupdate(empty)

reason (empty)

put	pected result	ata	ctual output	ass/fail
One or more empty	please fill this field”’	mpty startTime and y valid nfirmation ername	ease fill this field”’	Fail
Empty patientId or ctorId and all other valid lds	“Please enter patientId or or doctorId”	mpty patientId or ctorId and all valid lds	his field is required”	Fail
valid patientId doctorId nd all correct fields	“patientId or doctorId doesn’t match.”	valid patientId doctorId	atientId or doctorId esn’t match”	Fail
valid fields	“You have successfully dered a checkup.”	valid fields	“You have ccessfully Ordered a eckup.”	Pass

Table 7 : Test case specification for paying bill online

Name: pay bill online

Purpose: enable patients to paying bill online

Test Data= Payment method(valid, empty)

Credentials(valid, invalid, empty)

Input	Expected result	Data	Actual output	Pass/fail
All fields are empty	"fill all fields"	All fields are empty	"fill all fields"	Fail
Empty payment method And all other fields are valid	"Choose payment method"	Empty payment method And valid data for other fields	"Choose payment method"	Fail
Valid payment method and all other fields are valid	"Your information has been recorded. View payment history."	If payment method is specified and all are valid data for other fields	"Your information has been recorded. View payment history."	Pass

Invalid credentials and valid payment method	“Enter a valid credentials.”	When invalid credentials and valid payment method data	“Enter a valid credentials.”	Fail
Empty credentials and valid payment method	“Credentials cannot be empty!”	When user credentials are empty.	“Credentials cannot be empty!”	Fail

Table 8 : Test case specification for Search doctor

Name: **Search Doctor**				
Purpose: To let patient search a doctor				
Test Data Search (valid, invalid , empty)				
Input	**xpected result**	**Data**	**Actual output**	**Pass/fail**
One or more empty	“please fill this field””	The field is empty	“please fill this field””	Fail
Invalid doctor name	none	Invalid doctor name		Fail
Field is valid	“Doctor displayed”	“Valid Doctor name”	“Doctor displayed”	Pass

Table 9 : Test case specification for admin side

Name: **Admin Side**

Purpose:- the admin working

Test Data

patientId (valid, invalid , empty)

doctorId (valid, invalid , empty)

checkupdate(empty)

reason (empty)

Search (valid, invalid , empty)

Input	Expected result	Data	Actual output	Pass/fail
One or more empty	“please fill this field”’	The field is empty	“please fill this field”’	Fail
Invalid doctor name	Admin is redirected Login page	N/A	Admin is redirected to gin page	Pass
Admin without an count tries to login	Show error of "Wrong edential"	“Valid Doctor name”	Show error of "Wrong edential"	Pass

Valid account details	Admin is redirected to the Login page	Full Name, Email, Bio, Birthdate, Country, Password	Admin is redirected to the Login page	PASS
Valid login credentials	Admin is redirected to the Admin page	Username, Password	Admin is redirected to the Admin page	PASS
Valid doctor details	Doctor is displayed on the user's doctor page	Doctor Name, Specialization, Contact Information	Doctor is displayed on the user's doctor page	PASS
Valid facility details	Medical facility is displayed on the user's facility page	Facility Name, Location, Contact Information	Medical facility is displayed on the user's facility page	PASS
Valid category details	Treatment category is displayed on the user's category page	Category Name, Description	Treatment category is displayed on the user's category page	pass
Valid time slot details	New time slot is displayed on the user's scheduling page	Date, Time, Doctor, Duration	New time slot is displayed on the user's scheduling page	PASS
Existing time slot details	Time slot is removed from the user's scheduling page	Date, Time, Doctor	Time slot is removed from the user's scheduling page	PASS
Email with request details	Admin views appointment requests and responds accordingly	Patient Name, Date, Time, Doctor, Reason	Admin views appointment requests and responds accordingly	PASS

Reference

1. Jamil, M. A., Arif, M., Abubakar, N. S. A., & Ahmad, A. (2016). Software Testing Techniques: A Literature Review. *Research Gate*. https://doi.org/10.1109/ict4m.2016.045