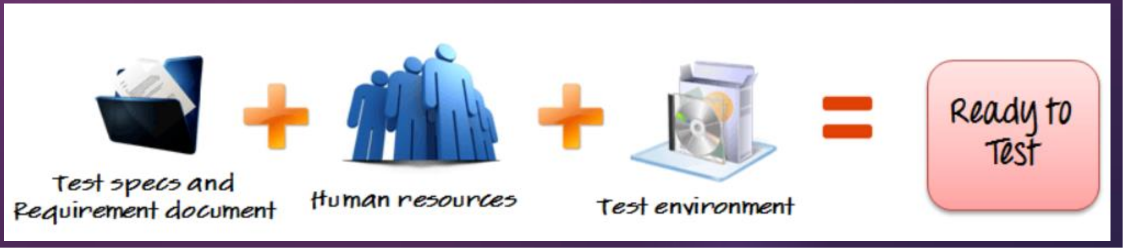
Software Testing

In support of the Sys Dev Project



Agenda

1. **RECAP:**
   - System Development Life Cycle (SDLC)
   - Software Testing

2. **Software Testing Documentation**
   - Test Plan
   - Test Scenarios
   - Test Cases

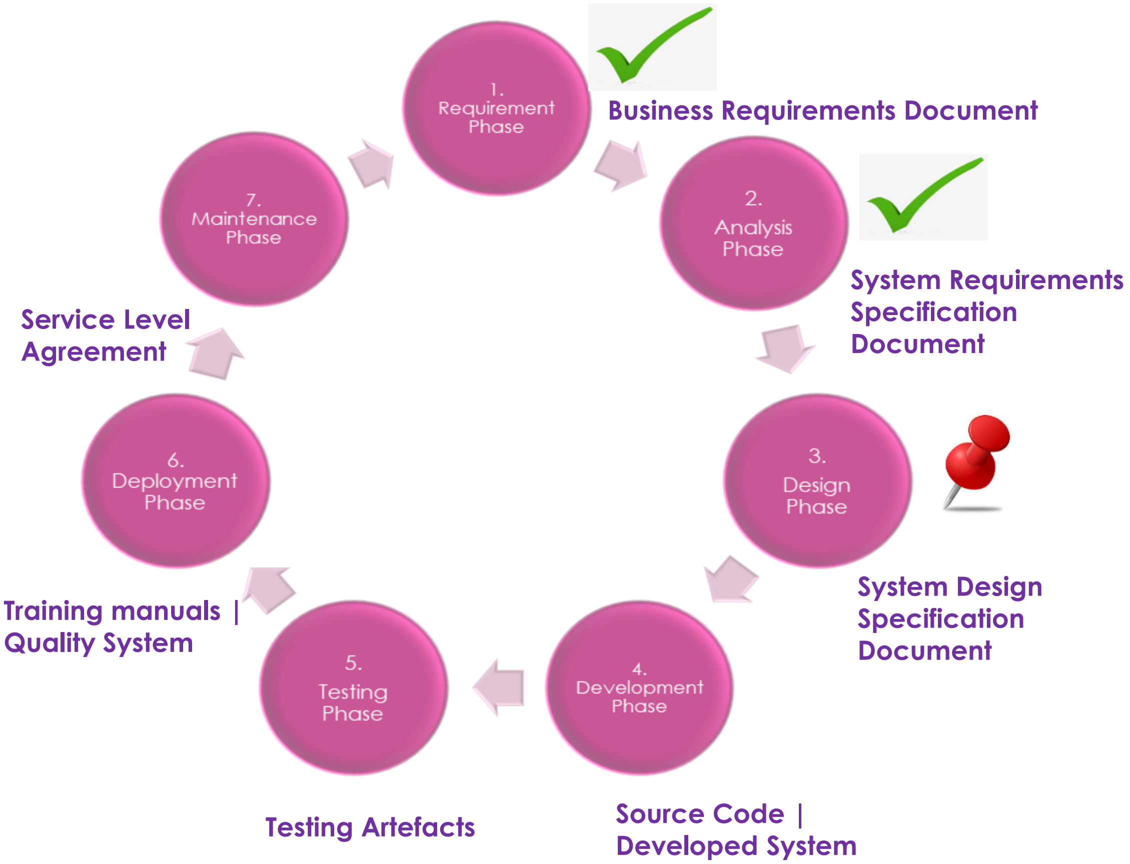
   Deliverables provided **BEFORE** the **Testing Phase**

   - Requirements Traceability Matrix

   Deliverable provided **AFTER** the **Testing Phase**

3. **Mind Map of an FNB System**

System Development Life Cycle (SDLC)



1. Requirement Phase ✓

Business Requirements Document

2. Analysis Phase ✓

System Requirements Specification Document

3. Design Phase

System Design Specification Document

4. Development Phase

Source Code | Developed System

5. Testing Phase

Testing Artefacts

6. Deployment Phase

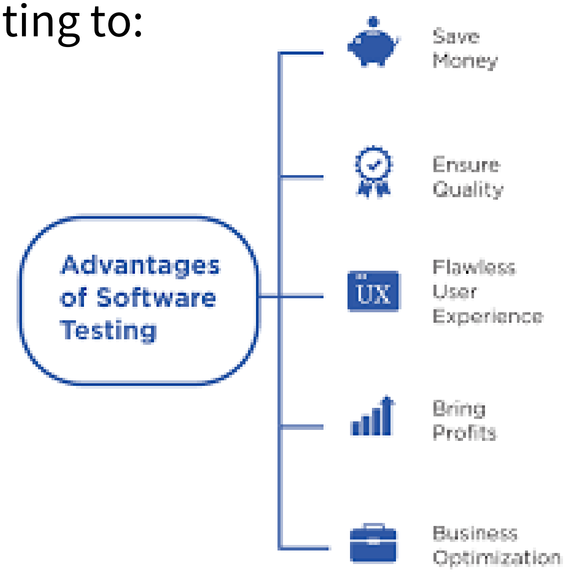
Training manuals | Quality System

7. Maintenance Phase

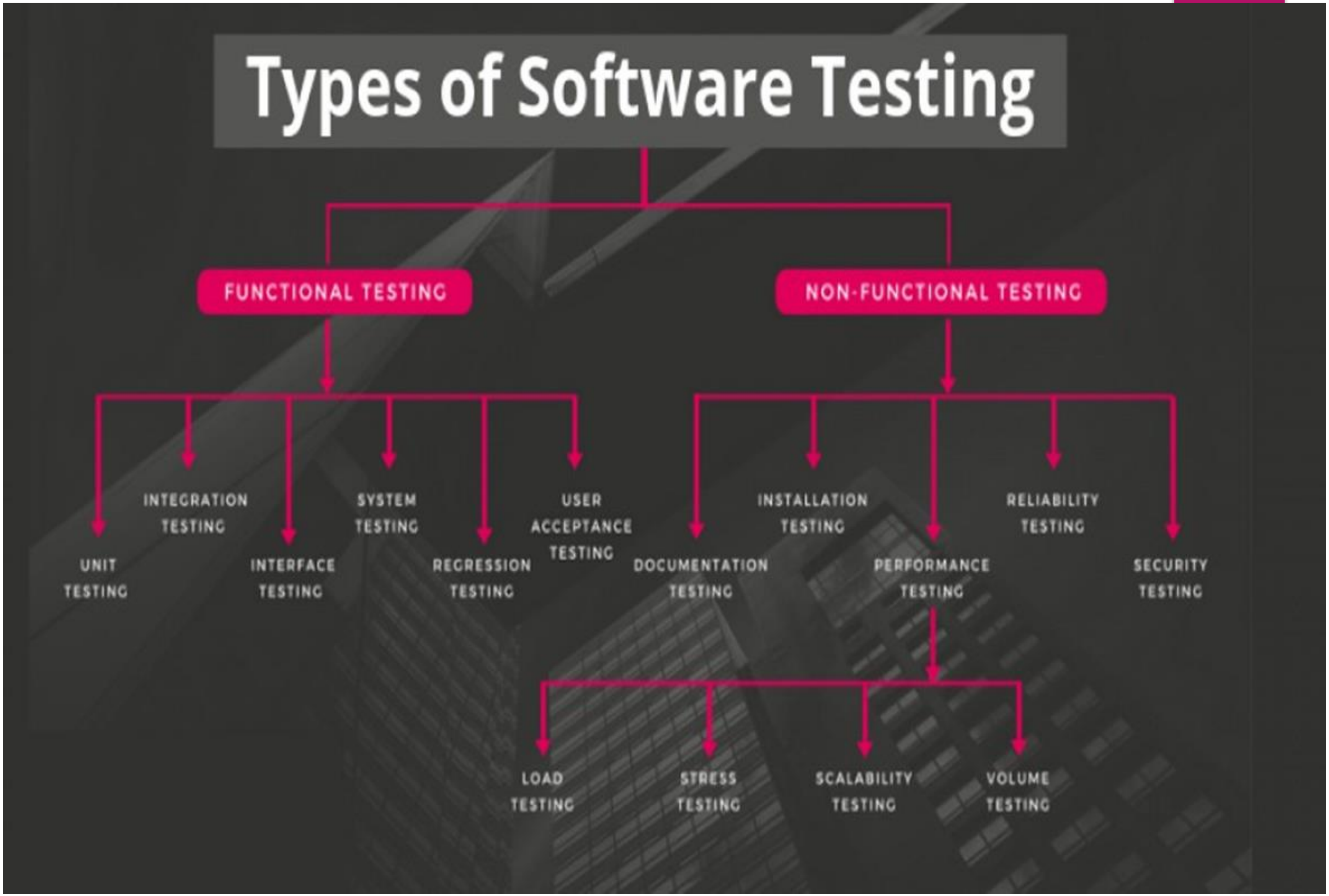
Service Level Agreement

Software Testing

- is a method to check whether the actual software product or system matches the expected requirements
- focuses on identifying errors, gaps, or missing requirements
- ensures that the software is defect free
- ensures that the software is verified and validated
- ensures reliability, security, and high performance, resulting to:
  - Time-saving
  - Cost effectiveness
  - Customer satisfaction
  - Preserves human lives



Advantages of Software Testing



Types of Software Testing

FUNCTIONAL TESTING

UNIT TESTING

INTEGRATION TESTING

INTERFACE TESTING

SYSTEM TESTING

REGRESSION TESTING

USER ACCEPTANCE TESTING

NON-FUNCTIONAL TESTING

DOCUMENTATION TESTING

INSTALLATION TESTING

PERFORMANCE TESTING

RELIABILITY TESTING

SECURITY TESTING

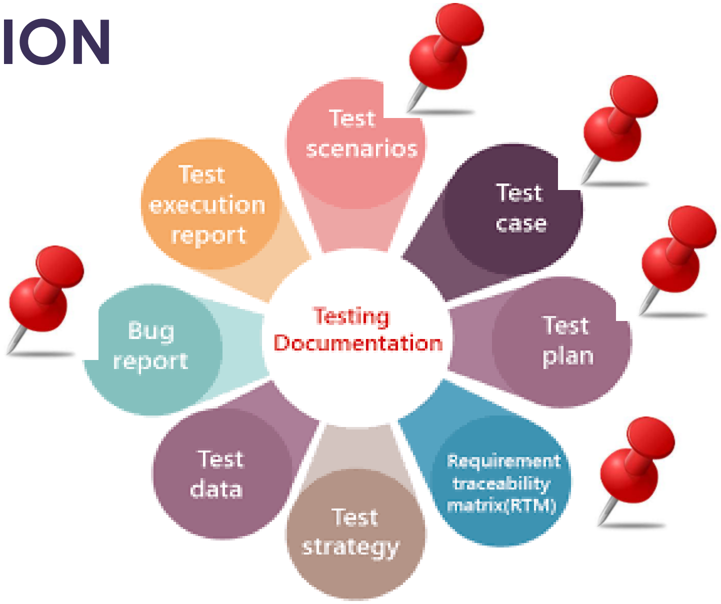
LOAD TESTING

STRESS TESTING

SCALABILITY TESTING

VOLUME TESTING

SOFTWARE TESTING DOCUMENTATION



Software Testing Documentation

Documents associated with software testing

Test Plan

Outlines the strategy to be implemented for testing a system

Test Scenario

- Documents a use case.
- Derived from BRS and SRS
- Ensures that every functionality a system offers is working as expected

Test Case

Steps and conditions with inputs that can be implemented at the testing time.

The following test cases should be done:

- Functional test cases
- Logical test cases
- Physical test cases
- User interface test cases

Traceability Matrix

Validates that all requirements are checked, and all the system functionalities are tested.

Test Plan

- It outlines the process of testing a system or software and should include:
  - Introduction, Test strategy, Objectives, Schedule/ Timescales, Environment, Acceptable criteria, Resources/People (testers) required to validate the system's quality, Risks, and mitigation measures, In-scope vs. out-of-scope etc.
- Acts as a blueprint to do software or system testing in a defined process
- Makes the testing process easy and organized
- Defines roles and responsibilities of testers
- Saves company costs
- Provides an overview of where to begin and end the work

Test Plan Example

Check the template for a full Test Plan

PROJECT TEST PLAN

INTRODUCTION AND STRATEGY

IN SCOPE

ENVIRONMENTS

PEOPLE

ACCEPTABLE CRITERIA

OUT OF SCOPE

RISKS AND MITIGATION MEASURES

TESTING REQUIREMENTS AND ASSUMPTIONS

TIMESCALES

DEFECT MANAGEMENT

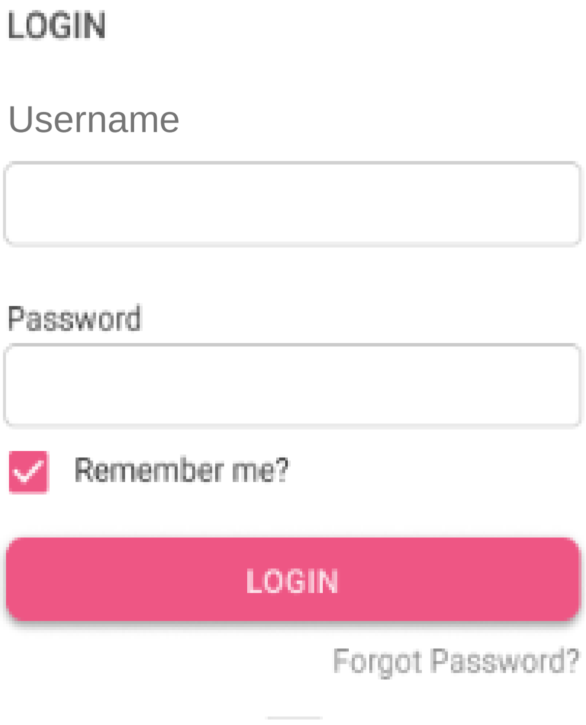
Test Cases

- Requires understanding of the system, requirements, and functionality
- Used to test the functionality of the system
- Can be created manually or automatically by using software testing tools such as JIRA or BugZilla
- Are developed based on the system requirements specification
- Are developed based on the agreed test case template and should include:
  - Test case ID
  - Test case description
  - Test Case Steps
  - Expected results
  - Actual results
- Must be clear, comprehensive and easy to understand

Test Cases Conti...

- Different types of test cases can be designed to test different aspects of the code, such as:
  - Functionality
  - Performance
  - Security
  - User interface
  - Integration
  - Databases
- In some cases, test cases may be designed to intentionally cause the code to fail.
  - **Example:** by providing incorrect input or attempt to access restricted areas.
  - This is done to verify that the code can handle such situations and respond appropriately rather than crashing or behaving unexpectedly

Test Case Example: User Login



Test Case Example: User Login

Check the template for more attributes

Test Case ID.	Test Case Description	Test Case Steps	Test Type	Input/Precondition	Expected Results	Actual Results	Status	Comment
UC01	User Login	Go to site: http://property24.com	Functional Test	Internet connection	User should log in to the website	As expected	Passed	
		Verify if there is a username field and if its editable	Unit Test		Username field is editable	Username field is editable	Passed	
		Verify if there is a password field and if its editable	Unit Test		Password field exist and its editable	Password field exist and its editable	Passed	
		Verify if there is a log in button	Unit Test		log in button to exist	log in exist but as a text field and not as a button	Failed	Fix log in to be a button
		Verify if there is a Forgot password button	Integration Test		User to be redirected to Forgot password page	Forgot password button is absent	Failed	Forgot password page under construction
		Verify whether the user can log in.	System Test	Username and password	User is allowed access to the home page	User is allowed access to the home page	Passed	
		Verify if a password verification works	Unit Test	Incorrect password	User unable to access the system	User unable to access the system	Passed	

Failed test steps go straight to the defect register

Defect Register: User Login

Test Case ID.	Test Case Description	Test Case Steps	Test Type	Input/Precondition	Expected Results	Actual Results	Status	Comment
UC01	User Login	Verify if there is a log in button	Unit Test		Log in button to exist	Log in exist but as a text field and not as a button	Failed	Fix log in to be a button
		Verify if there is a Forgot password button	Integration Test		User to be redirected to Forgot password page	Forgot password button is absent	Failed	Forgot password page under construction

Steps to be taken for all failed steps:

- System Requirements Specification documents
  - ✓ check the functional spec
- Align test case to the requirements and design
- **If it passes**, it goes back to the test case register

Test Case Example: Browser

- Test the system on different browsers and operating systems to ensure it is compatible with as many users as possible
- Decide on which browser your system or app supports
- **Common browsers** include the following:

Internet Explorer Firefox Chrome Safari Opera

TEST steps:

- Check if the website/app loads on a browser
- Check if the elements (i.e. buttons, forms, menus) are visible
- Check if the elements are clickable on a page
- Check if the webpage takes a longer time to load on the browser

Test Scenario

Test scenarios are created to:

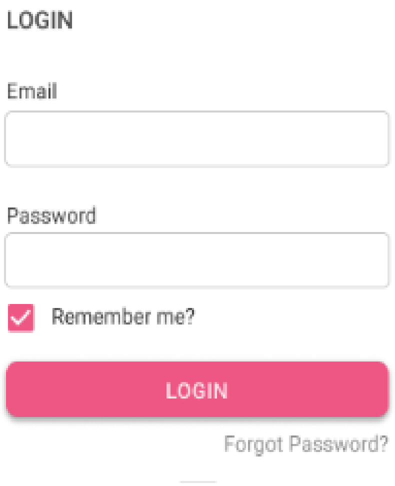
- validate that the system is working appropriately for each use case
- improve user experience
- identify the most important end-to-end product
- ensure the completion of test coverage

Who writes the Test scenarios?

- Testers
- Business analysts
- Test Leads
- **However**, mostly dependent on the project or organisation

Test Scenario Example: User Login

Test Scenario: Check the Login functionality


LOGIN

Email

Password

☑ Remember me?

Ask the following questions in this scenario:

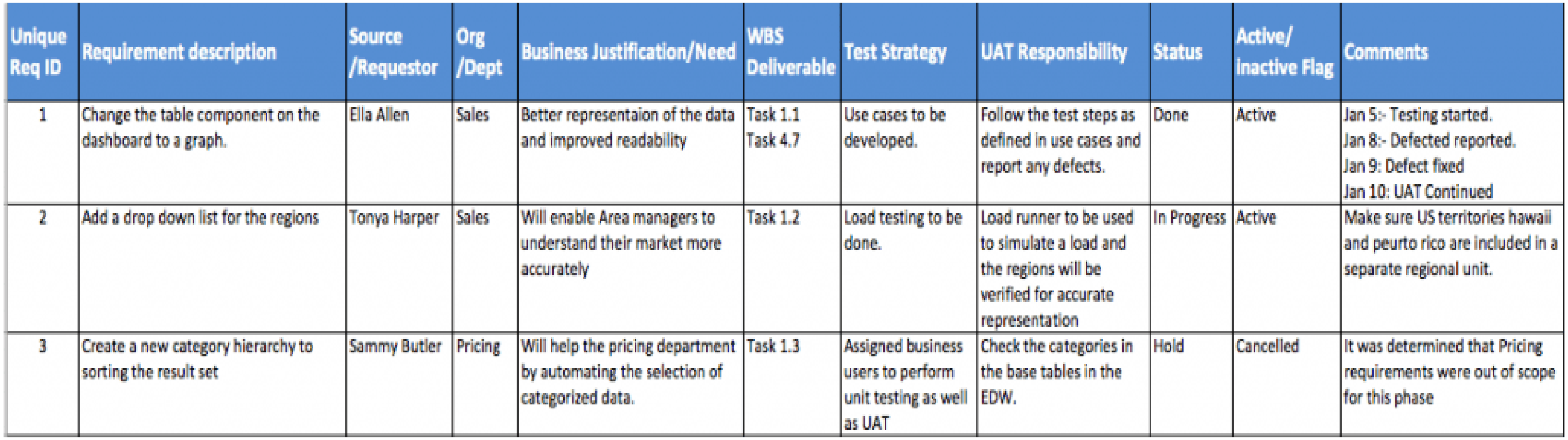
- Can the user log in with a valid email and password?
- What happens when an invalid email and valid password are entered?
- What happens when a **valid email and an invalid password** are entered?
- What happens when an **invalid email and invalid password** are entered?
- What happens when the **email and password fields are blank and the Login button** is clicked?
- Is there a Forgot password? Is the option working as expected?
- Is Remember me checkbox functioning as expected?

Requirements Traceability Matrix (RTM)

- Used to demonstrate that the requirements have been fulfilled
- Maps and traces user requirements with test cases
- Testers can write test cases from the business requirements
- Delivered at the end of the software development life cycle
- *Simple terms: RTM ensures that all the requirements are met*

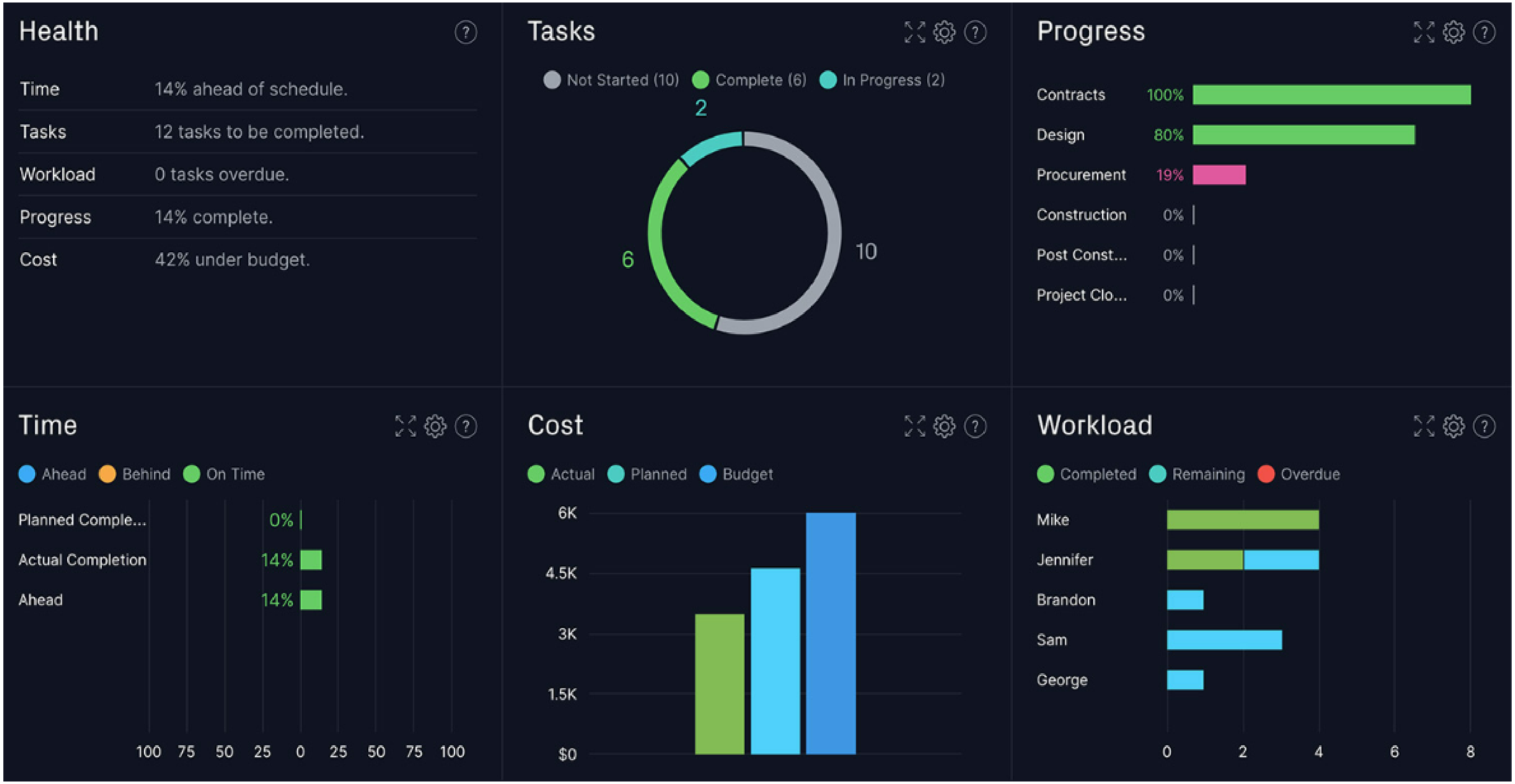
- **There are three types of requirements traceability**
  - **Forward traceability** – used to check if the system is meeting the requirements
  - **Backward traceability** – checks if the system is on track
  - **Bi-directional traceability** – maps requirements to test cases and test cases to the requirements

Requirements Traceability Matrix Example



Unique Req ID	Requirement description	Source /Requestor	Org /Dept	Business Justification/Need	WBS Deliverable	Test Strategy	UAT Responsibility	Status	Active/ Inactive Flag	Comments
1	Change the table component on the dashboard to a graph.	Ella Allen	Sales	Better representaion of the data and improved readability	Task 1.1 Task 4.7	Use cases to be developed.	Follow the test steps as defined in use cases and report any defects.	Done	Active	Jan 5:- Testing started. Jan 8:- Defected reported. Jan 9: Defect fixed Jan 10: UAT Continued
2	Add a drop down list for the regions	Tonya Harper	Sales	Will enable Area managers to understand their market more accurately	Task 1.2	Load testing to be done.	Load runner to be used to simulate a load and the regions will be verified for accurate representation	In Progress	Active	Make sure US territories hawaii and peurto rico are included in a separate regional unit.
3	Create a new category hierarchy to sorting the result set	Sammy Butler	Pricing	Will help the pricing department by automating the selection of categorized data.	Task 1.3	Assigned business users to perform unit testing as well as UAT	Check the categories in the base tables in the EDW.	Hold	Cancelled	It was determined that Pricing requirements were out of scope for this phase

Requirements Traceability Matrix Example (Visual presentation)

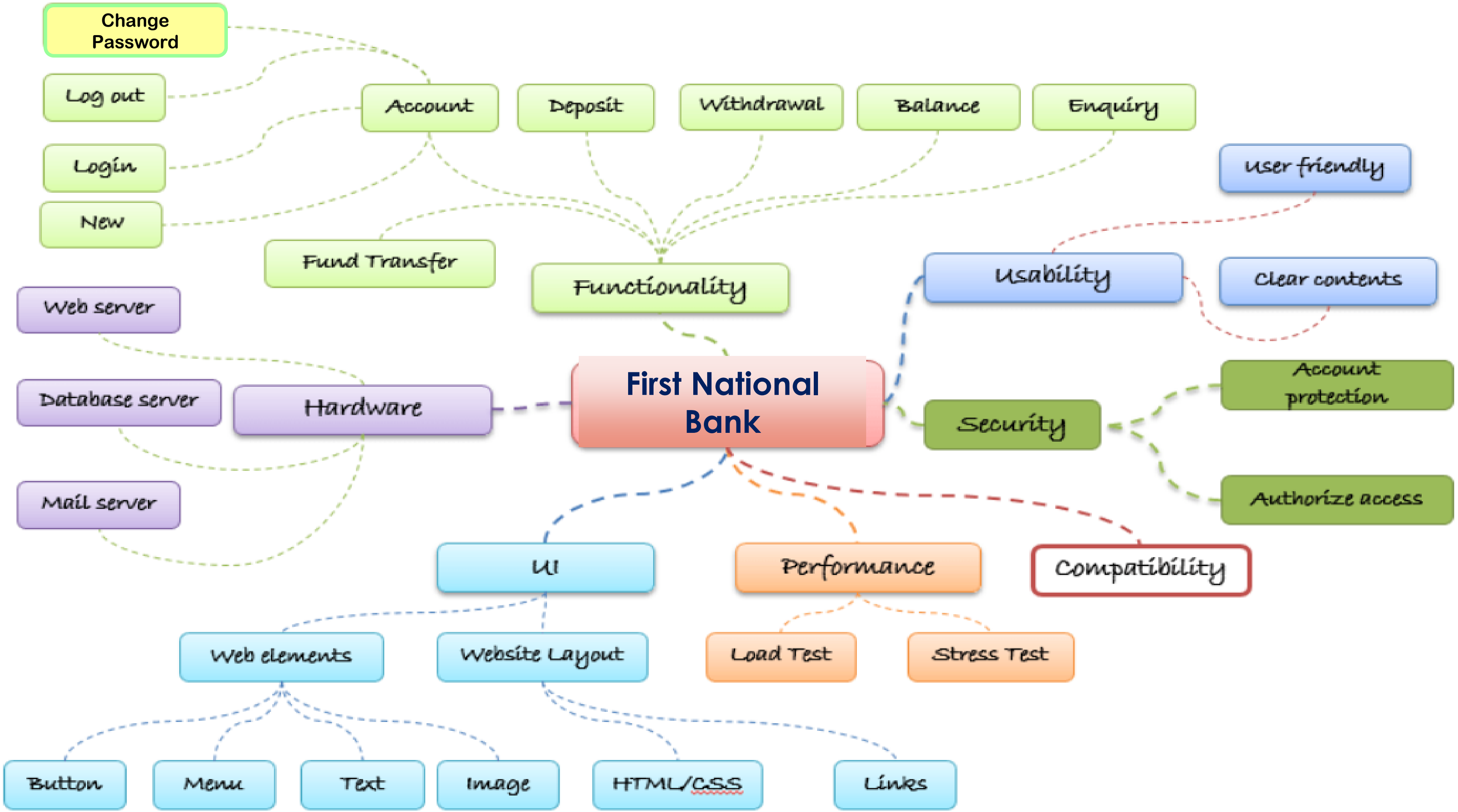


Health

Time	14% ahead of schedule.
Tasks	12 tasks to be completed.
Workload	0 tasks overdue.
Progress	14% complete.
Cost	42% under budget.

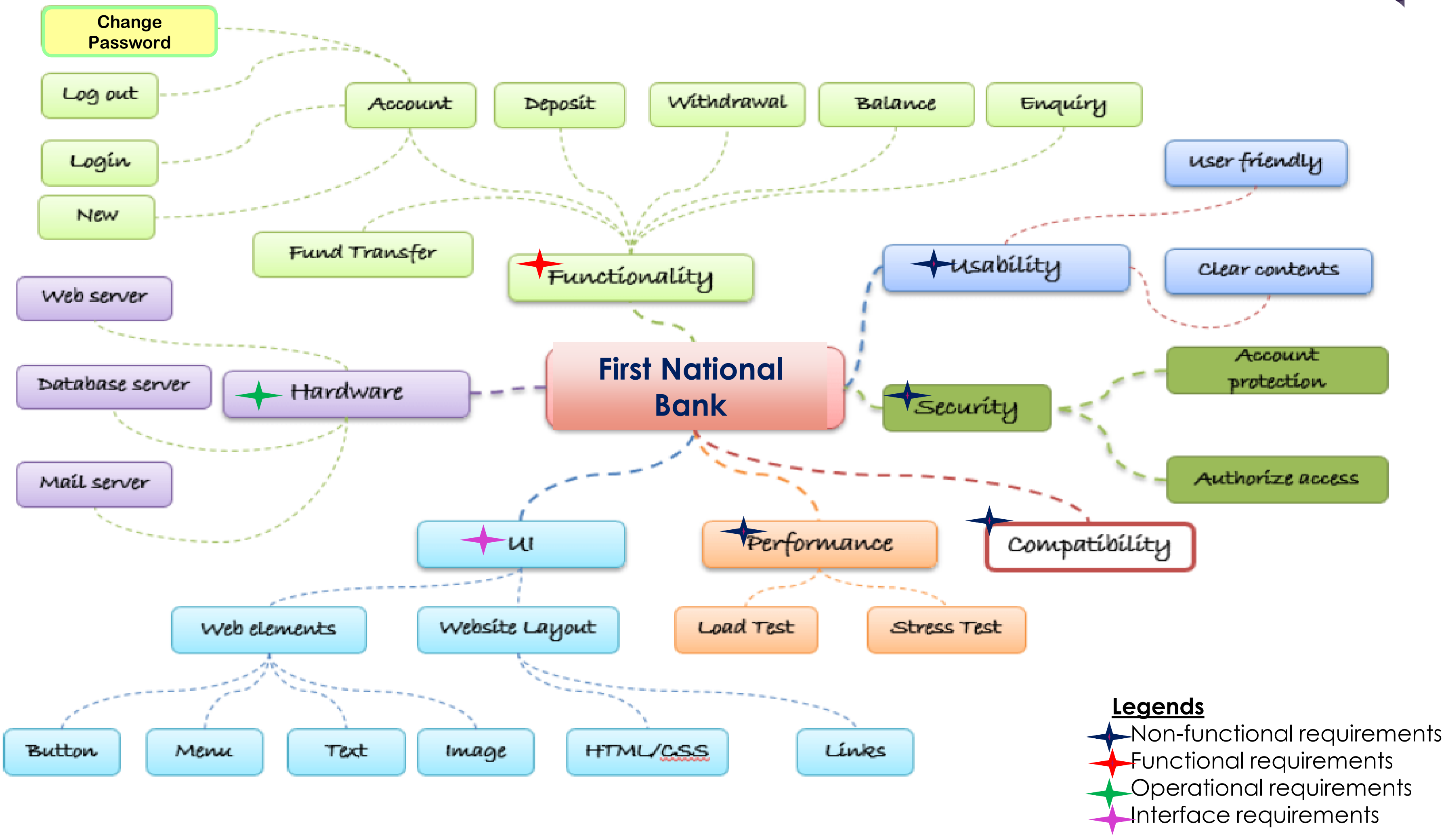
Mind map of a system

Mind Map of an FNB System



Suggestion: consider designing something similar to this for your project

Mind Map of an FNB System **with Legends**



First National Bank

Functionality

Change Password

Log out

Login

New

Account

Deposit

Withdrawal

Balance

Enquiry

Fund Transfer

Usability

User friendly

Clear contents

Security

Account protection

Authorize access

Hardware

Web server

Database server

Mail server

UI

Web elements

Website Layout

Button

Menu

Text

Image

HTML/CSS

Links

Performance

Load Test

Stress Test

Compatibility

Legends

✦ Non-functional requirements

✦ Functional requirements

✦ Operational requirements

✦ Interface requirements

Suggestion: consider designing something similar to this for your project

Students Expectations

- **To be able to create the following:**
  - Test Plan
  - Test Scenarios
  - Test Case document for all major use cases
  - Requirements Traceability Matrix

Thank you, and all the best for this semester ☺

End

THANK YOU