Application Requirements

1. The core entities are **Bugs** and **Users**.
2. The following information is required for all users, **email**, **password**, **full** **name**, **given** **name**, **family** **name**, and **role**.
3. Users can log into the system using their **email address** and **password**.
4. User can **recover the password** using forgot password feature.
5. Users can have different **roles** such as **Developer**, **Business Analyst**, **Quality Analyst**, **Product Manager**, **and Technical Manager**.
6. Any user can **create** a bug, and all bugs must contains following information:
   1. Title (must be unique)
   2. Description
   3. Department : Logistics | Marketing | Operations | IT
   4. Type of Issue : BUG | QUESTION | CHANGE |
   5. Steps to reproduce (store as a string, not a separate entity)
   6. Expected Behavior
   7. Author of bug
   8. Date and time of creation
   9. Date and time of completion
   10. Priority
   11. Status
   12. Screenshots
7. Any user can **edit** the bugs that they created.
8. Bugs can be **assigned** to either Developers, or Business Analysts, or Quality Analysts.
9. These users can **modify** the bugs that they are assigned to. But not bugs that they did not create or are not assigned to.
10. All edits must be **tracked** with the date & time the change was made, the user who made the change, and what was changed.
11. Anyone can add **comments** to a bug, regardless of who created it or who is assigned to. These comments must include the author, date, and time of creation. Bugs can have multiple comments.
12. Comments cannot be edited.
13. **Business Analysts** can classify bugs are **approved**/**unapproved**/**duplicate**. By default bugs must be **unclassified**.
14. **Quality Analysts** can add **test cases** to bugs and mark them as **passed**/**failed**. Bugs can have multiple test cases.
15. **Developers** can track the amount of **time (in hours)** that they spent on a bug. Developers can submit hours multiple times.
16. **Developers** can note what **version** of the software the fix will be released in and **when** the fix was made to the note.
17. **Business Analysts** can **close** bugs when they are released to customers, but all of the previous information must be retained indefinitely.

**Exercise 1: Create ERD**

1. All database entities (tables and sub-tables)
2. All relationships between entities
3. The cardinality of each relationship (one-to-many / many-to-one)
4. The properties of all entities
5. The data types of all properties
6. The Primary key in every table must be named **<*name or table*>Id** and listed at the top.

**Exercise 2: Implement Database**

**Exercise 3: Query Database**

Write and test queries for each of the prompts below

Save these queries to a file named “queries.sql”

Some of these queries may require use of an aggregate pipeline.

1. List all of the bugs that your user created.
2. List all of the bugs that you are assigned to.
3. List all of the bugs that you created or are assigned to.
4. List all of the bugs that are unclassified.
5. List all of the bugs that have been approved by a Business Analyst, but are not closed.
6. List all of the bugs that fixed, but are not closed.
7. List all of the bugs that do not have steps to reproduce.
8. List all of the bugs that do not have test cases.
9. List all of the bugs that do not have hours/work/labor recorded.
10. List all of the bugs, with the number of comments for each bug.
    1. Do not list the same bug multiple times
    2. Include bugs with no comments
11. List all of the bugs, with the total number of hours worked on each bug.
12. List all of the bugs, with failing test cases.

Customer Dashboard

**Filters:**

1. Filter by Priority : MEDIUM | LOW | CRITICLE | HIGH
2. Filter by Point of Contact (Task Creator email)
3. Filter by Status : OPEN | IN-PROGRESS | DONE

**Bug Table must contains**

1. Title
2. Priority
3. Description
4. Date Created
5. Date Resolved
6. Status
7. Point of Contact
8. Tags
9. Current Assignee
10. Actions : View (Bug History)

**Create Bug**

1. Label
2. Project Dropdown
3. Title
4. Priority
5. Description
6. Steps to reproduce

Admin Dashboard

**Filters:**

1. Filter by Priority : MEDIUM | LOW | CRITICLE | HIGH
2. Filter by Point of Contact (Task Creator email)
3. Filter by Status : OPEN | IN-PROGRESS | DONE

**Bug Table must contains**

1. Title
2. Priority
3. Description
4. Date Created
5. Date Resolved
6. Status
7. Point of Contact
8. Tags
9. Current Assignee
10. Actions : View (Bug History) | Edit | Delete | Mark Duplicate

**Create Bug**

1. Label
2. Project Dropdown
3. Title
4. Priority
5. Description
6. Steps to reproduce

Create Priority

1. Priority Name

List of Categories

Create User

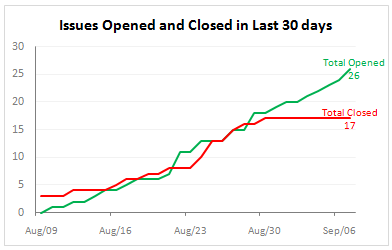
1. Full Name
2. Role
3. Email Address
4. Password
5. Confirm Password

List of Users

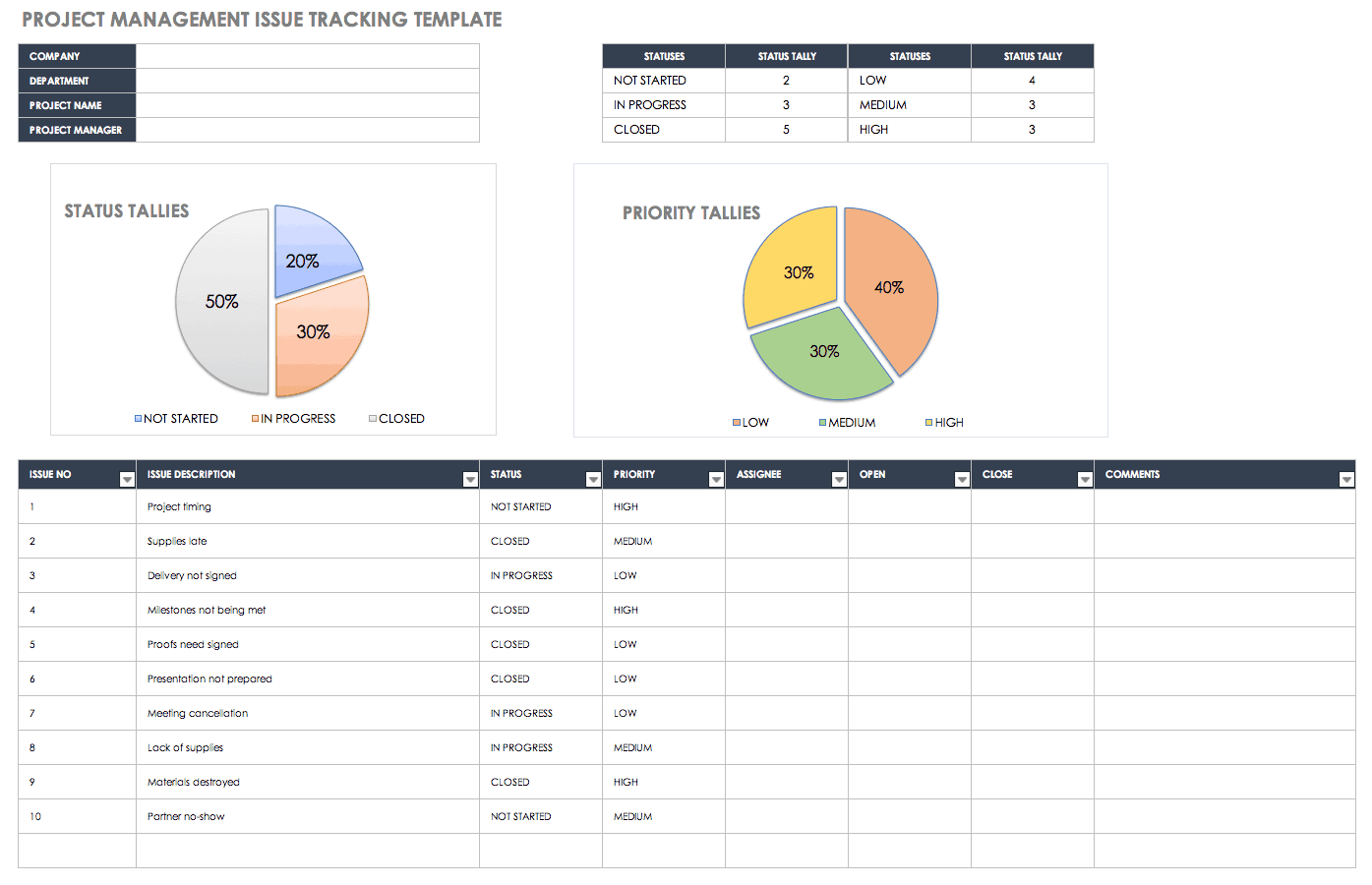
1. Name
2. Email
3. Role
4. Actions : Edit | Delete

Reports

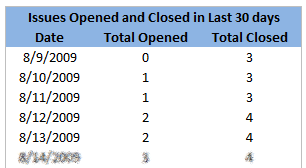
**Open vs. closed issues chart**



**Status & Priority Tallies**



**Issue counts for the last 30 days**



**Dashboard**

