Contents

**1. Introduction:**

* 1. Purpose & Scope
  2. Definitions, acronyms, abbreviations
  3. Overview

# 2. Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

# 3. Specific Requirements

**3.1 External Interface Requirements**

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

**3.2 Functional Requirements**

3.2.1 Login

3.2.2 Registration

3.2.3 View/Filter Bug Reports

3.2.4 User Profile

3.2.5 Staff Dashboard

**3.4 Design Constraint**

3.4.1 Standards compliance

3.4.1 Standards compliance

**3.5 Software System Attributes**

3.5.1 Reliability

3.5.2 Availability

3.5.3 Security

3.5.4 Maintainability

3.5.5 Portability

**4. Design**

**5.Implementation**

**6. Testing**

**7. Conclusion**

1.1 Purpose & Scope:

**Goals & Objectives:**

* Developing a flawless system
* Track all the defects or bugs
* Avoid Miscommunication/Time Consumption
* History/Records

**1.2 Definition:**

**What is Bug Tracking System?**

Bug-Tracking System is an ideal solution to track the bugs of a product, solution or an application. Bug Tacking System allows individual or groups of developers to keep track of outstanding bugs in their product effectively. This can also be called as Defect Tracking System.

**Roles in Bug Tracking System**

* User
* Administrator
* Developer
* Testing

1.3 Overview of The System

The process of our tracking system is:

* User submits bug reports
* Status is Changed to: **NEW**
* Admin assign bug report to developer
* Change Status To: **ASSIGNED**
* Developer fixes issues regarding report then

**IF** Bug Is Resolved Then

* STATUS Changed to: **RESOLVED**

**Else**

* Status Is Changed To: **REASSIGNED**
* Tester will then test reports solved by developer for any remaining anomalies

**IF** found no bug or further issues then

* Status is changed to: **CLOSED**

**Else**

* Status is changed to: **REASSIGNED**

2. Overall Description

2.1 Product Perspective

Database of Bug Tracking system stores the following information.

* + - Administrator, User, Developer, Tester's Information
    - Account information
    - Project Information
    - Project State & Priority

2.2 PRODUCT FEATURES:

**Features:**

Query Bug List to any depth

* + Reporting in more comprehensive way
  + User Accounts to control the access and maintain security Simple Status & Resolutions
  + Multi-level Priorities & Severities
  + Attachments & Additional Comments for more information
  + Robust database back-end

2.3 User characteristics

1. User:

* User must Enter Proper Information About the Project & Must Attach the Project Files
* User Can View the Status of Bug Report & Can Edit His / Her Bug Report
* Will get the Fixed Project on Given Date/time

2 . Administrator:

* + Can Assign Submitted Bug Report to Developer
  + Can Change Status of Bug Report
  + Can Edit Bug Report
  + Can Delete Bug Report
  + Can Add Accounts (User, Developer, Tester)

3. Developer:

* Can View/Search Bug Reports
* Can Re-Assign Bug To any other Developer
* Can Assign Fixed Project to Tester
* Can Change Status of Bug Report
* Should Fix Assigned Bugs
* Will get Email when Assigned

4. Tester:

* Should Test Assigned Bugs
* Can View/Search Bug Reports
* Can Change Status of Bug Report
* Will get Email when Assigned

2.4 OPERATING ENVIRONMENT:

Operating environment for Bug Tracking System is as listed below.

* + Centralized database
  + client/server system
  + Operating system: Windows.
  + database: MySQL + database
  + platform: ASP.NET

**2.5 Design & Implementation Constraints:**

* + **Design Flexibility:**

We have designed our system keeping it flexibility in mind for e.g. For client side we used:

Bootstrap for making design flexible and responsive for any platform (OS) e.g. (Windows, MAC,UBUNTU, Android) etc.

Our design is flexible to any further change for example change in libraries due to evolvement.

e.g. change in jQuery libraries, etc.

Also we have taken care of code design so it would be well understood and be clear for any other

developer working on it etc. Taking Care of all users we have not included the .Most newer designs which are incompatible for Users who still are using older version of browsers As it is evolutionary system it needs its design (User

Interface) to be changed over time that’s why we have made it UI design flexible to be changed as Users’ taste changes.

* **Language requirements:**

We have used Globalization with localization to facilitate

users with their personal language

For e.g. people living in USA prefer the language of the site

to be English whereas Russian people would prefer

Russian language for this reason we have to give users a

choice for choosing their language according to their

preference by using drop down menus

Our project is translated into 3 languages

Which are:

* English
* Urdu
* Chinese

While most of the thing have been translated but some of them are still

not been translated due to it has been restricted due to database

entries

These things of strings are of database and are not recommended to

be translated as a purpose of centralize data understanding.

**2.6 Assumptions & Dependencies:**

* Time Dependencies
* Hardware Dependencies (Server, Network , PC)
* External Dependencies(Email)

**3. Specific Requirements**

3.1 External Interfaces Requirements

**3.1.1 User Interface:**

This system is purely user friendly for administrator and developer. Following screens has been provided

A login screen for entering the username and password, so that the authorized user can have an access without any problems.

There will be a screen which will be displaying the major tasks that the system will be performing i.e. organization, bugs, reports, security.

Each major tasks has some tasks inside that as for example; in organization, view projects and view members are present, in bugs; view projects and view bugs are present.

All the major tasks mentioned above will have their separate forms and will perform the desired actions.

**3.1.2 Hardware Interface:**

As the Bug Tracking System is a Web Based

Application so it requires only Supporting Internet Browser can be run on almost every Computer.

**3.1.3 Software Interface:**

* + Operating System: Windows XP/7/10 or Linux or Mac OS
  + Programming Language: ASP .NET
  + User Interface: Html, CSS
  + Client Side Scripting: Java Script (JQuery) Database: My SQL
  + IDE: Microsoft Visual Studio 2013 or 2015
  + Server Deployment: <https://www.VirtualBreez.com>

**3.1.4 Communication Interface:**

All HTML pages will be transmitted with standard

HTTP protocol.

The user’s request will move to the server and will be notified to administrator managing the reports of all users.

A notification would be sent by email, from the system to the user in response using SMTP protocol.

**3.2 Functional Requirements**

**3.2.1 Mode 1**

**Title**: New User Can Register Account

**DESCP**: The user should be able to register through Registration form. The user must provide user-name, password, DOB. & e-mail address.

The user can choose to provide a regularly used phone number. All the registered information will be stored in the Database.

**3.2.2 Mode 2**

**Title**: User/Staff can Login to the System

**DESCP**: Given that a user has registered, then the user should be able to log in to the Bug Tracking System and after login the user will be redirected to the Bug Tracking System.

**3.2.3 Mode 3**

**Title**: Staff will be able to Search Submitted Bug reports.

**DESCP**:

Developer, Tester & Administrator have access to Search all Bug Reports Through Bug Report Title.

**3.2.4 Mode 4**

**Title**: Staff Dashboard

**DESCP**:

After login a Dashboard will be Displayed on Staff’s Screen Through which he/she can view Assigned reports .

**3.2.5 Mode 5**

**Title**:

**DESCP**: View Submitted Reports

Developer, Tester & Administrator Can View Reports Submitted By Any User .

**3.2.6 Mode 6**

**Title**: View Solved Reports

**DESCP**:

Staff can View All Solved Reports.

**3.2.7 Mode 7**

**Title**: User Profile

**DESCP**:

User / Staff Can View Anyone’s Profile .

**3.3 Design Constraints**

**3.3.1 Standard Compliance**

IEEE Standards 830-1993 for SRS

**3.3.2 Hardware Limitations Fault Tolerance:**

System should be able to tolerate some faults in system. Machines:

OS should be popular e.g. Windows, Mac, Android, Ubuntu etc. or having web browsers.

Some machines or software may need to tolerate faults.

Little effect of UI as hardware changes.

Restrictions are imposed in design due to multiple platform application.

**3.4 Software System Attributes**

**3.4.1 Reliability**

* **Replication of database**

The system shall provide for replication of databases to off-site storage locations.

* **Raid 3 Stripping**

The system shall provide RAID 3 Disk Stripping on all database storage disks.

* **Providing 24/7 availability**

The system shall provide a contractual agreement with an internet service provider who can provide 99.999% availability through their network facilities onto the internet.

**3.4.2 Security Login panels for everyone**

* Automatic logout / end session process
* Deleting confidential cookies

The system shall not leave any cookies on the customer’s computer containing the user’s password.

**3.4.3 Performance:**

* Web Based
* Initial load time dependent on Internet Service
* Hardware Configuration dependability

**3.4.4 Portability:**

* Environment having web browsers:

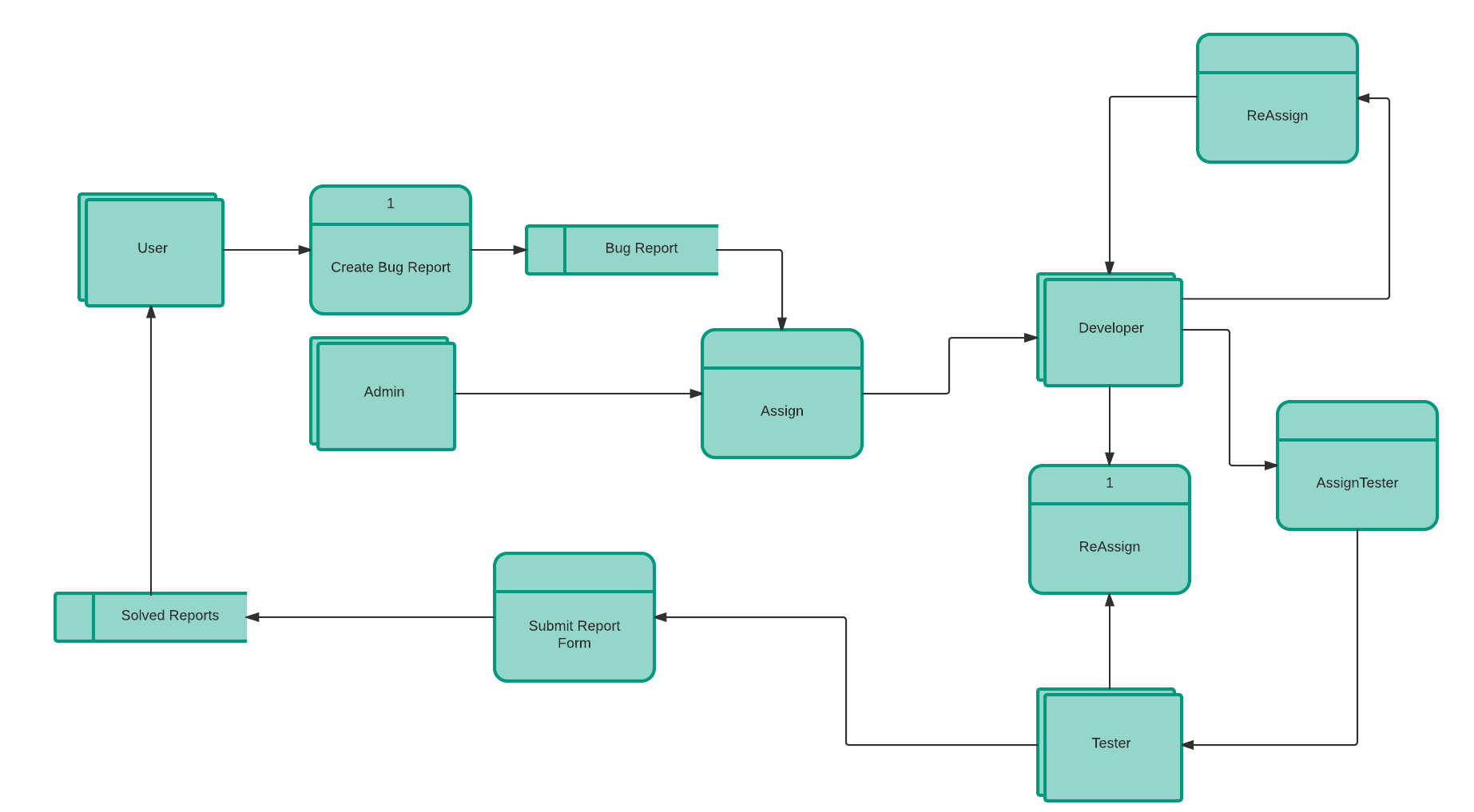
As it is web based so it should be accessible to/from any system hardware or software environment containing web browsers.

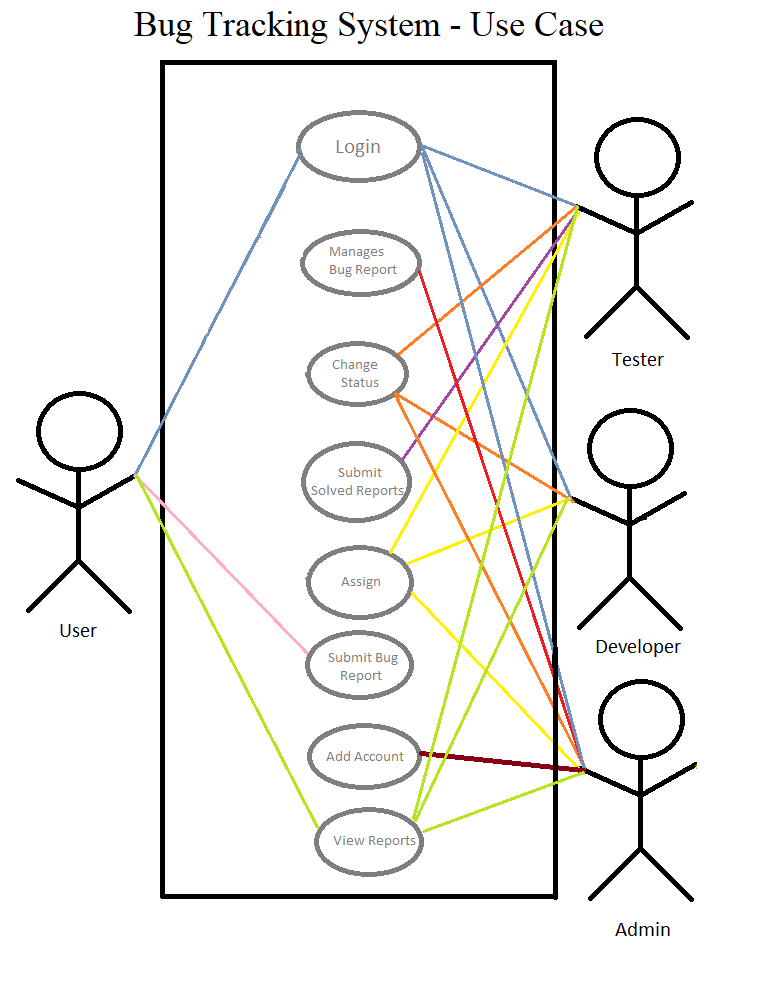
**3.4.5 Maintainability:**

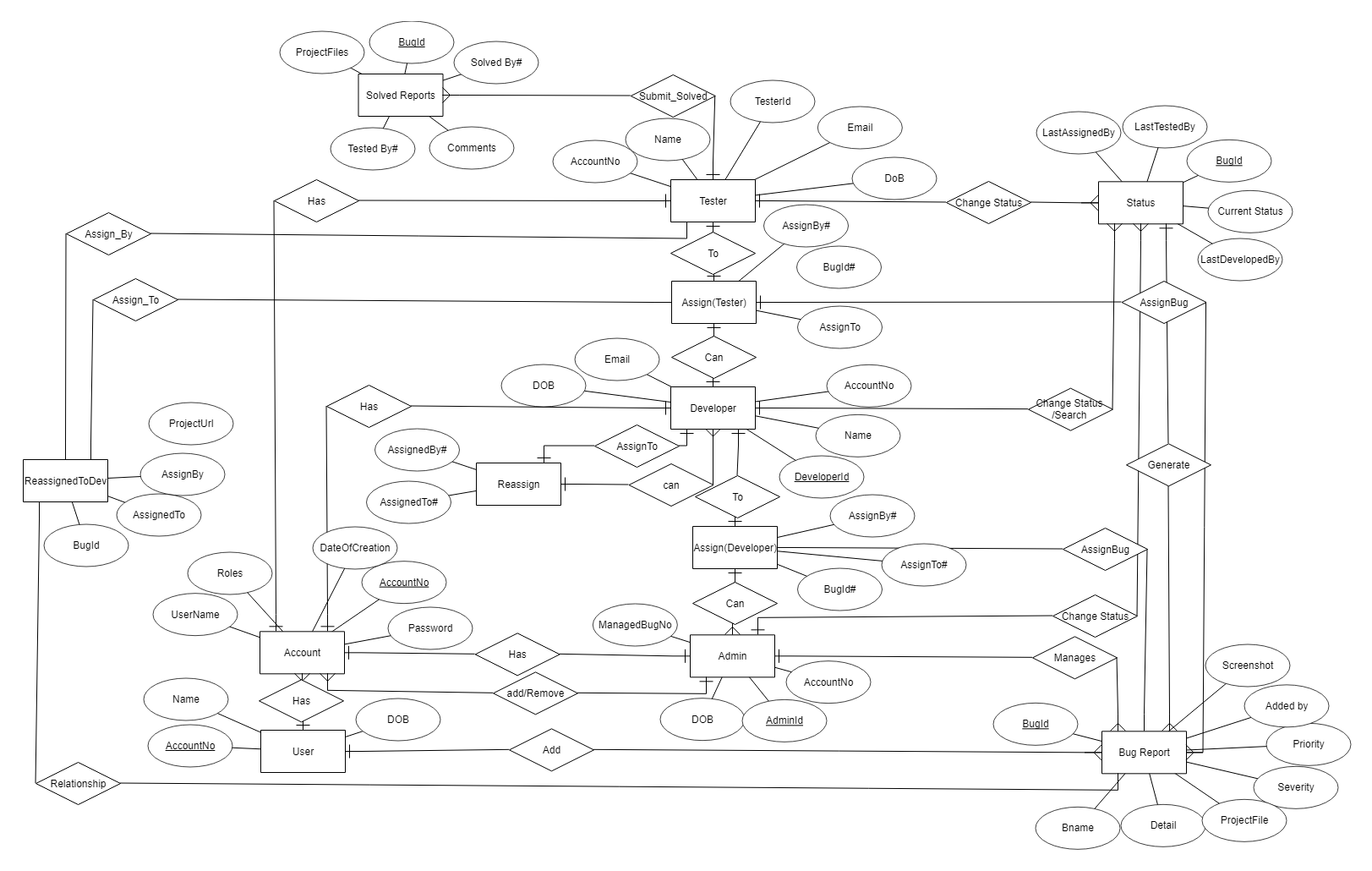
* Line of codes should be minimized and code should be divided into functions
* Possibly less function size
* Cloning of objects
* Not violating coding rules
* Dynamic array size
* Coding standards e.g. preferred letter case or other for function names etc.
* Dynamic user interface
* Modern libraries
* Improving security as increasing vulnerability with time

# 5.Design

# Bug Tracking System – Data Flow:



****



# 6. Development

We have divided Our Project into Modules That are explained Below

**Unit 1**

**New User Can Register Account**

**Explain**: The user should be able to register through Registration form. The user must provide user-name, password, DOB. & e-mail address.

The user can choose to provide a regularly used phone number. All the registered information will be stored in the Database.

**Unit 2**

**User/Staff can Login to the System**

**Explain**: Given that a user has registered, then the user should be able to log in to the Bug Tracking System and after login the user will be redirected to the Bug Tracking System.

**Unit 3**

**Staff will be able to Search Submitted Bug reports.**

**Explain**:

Developer, Tester & Administrator have access to Search all Bug Reports Through Bug Report Title.

**Unit 4**

**Staff Dashboard**

**Explain**:

After login a Dashboard will be Displayed on Staff’s Screen Through which he/she can view Assigned reports .

**Unit 5**

**View Submitted Reports**

**Explain**:

Developer, Tester & Administrator Can View Reports Submitted By Any User .

**Unit 6**

**View Solved Reports**

**Explain**:

Staff can View All Solved Reports.

**Unit 7**

**User Profile**

**Explain**:

User / Staff Can View Anyone’s Profile .

# Testing:

* Unit Testing

Since we have divided BTS into different Units . All Units are individually Tested.

* Integration of units:  
  After Unit Testing all Units Are Integrated
* Integration Testing

When Integration is Done the Whole Integrated unit is Tested

* Fixing issues after each testing phase

All Problems occurred during Testing Phase are Fixed.

# Conclusion

User comes to the search engine and makes a query, typically by giving key words, the engine looks up the index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text.  
Most search engines employ methods to rank the results to provide the “best” results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another.  
Search engine is technically the software and algorithms used to perform a search, the term have become synonymous with the website itself.