This is the requirement document for a C# based application which is a bug tracker. This new system will allow businesses and employees to keep track of bugs within the development cycle of new products along with the upkeep of old products.

This project is bug tracker without internet connectivity for use in software development environments both commercial and recreational.

The purpose of this product is to make the development and testing processes easier. Due to the fast nature of software development cycles, it can be difficult to keep track of software errors and communicate to other team members notifying them of errors within the code. The system is based upon a relational database system which will store user created logs containing data about how the errors occur, metadata about when and where the error occurred and who created the log along with many other features.

This project will support both companies and users of the software in order to create a cohesive system. The three main users of this software are; companies, employees, and general users.

Each user has slightly different requirements for the system however the general framework is the same.

The bug tracker will store the following information

Bug log:

* Bug description
* Unique ID for bugs
* Status of bug
* Application and version the bug occurs in
* Classifiers or category of the bug
* Confirmation of the bug
* Who reported the bug
* When was the bug first discovered
* When the log created
* When was the bug resolved
* Developer comments

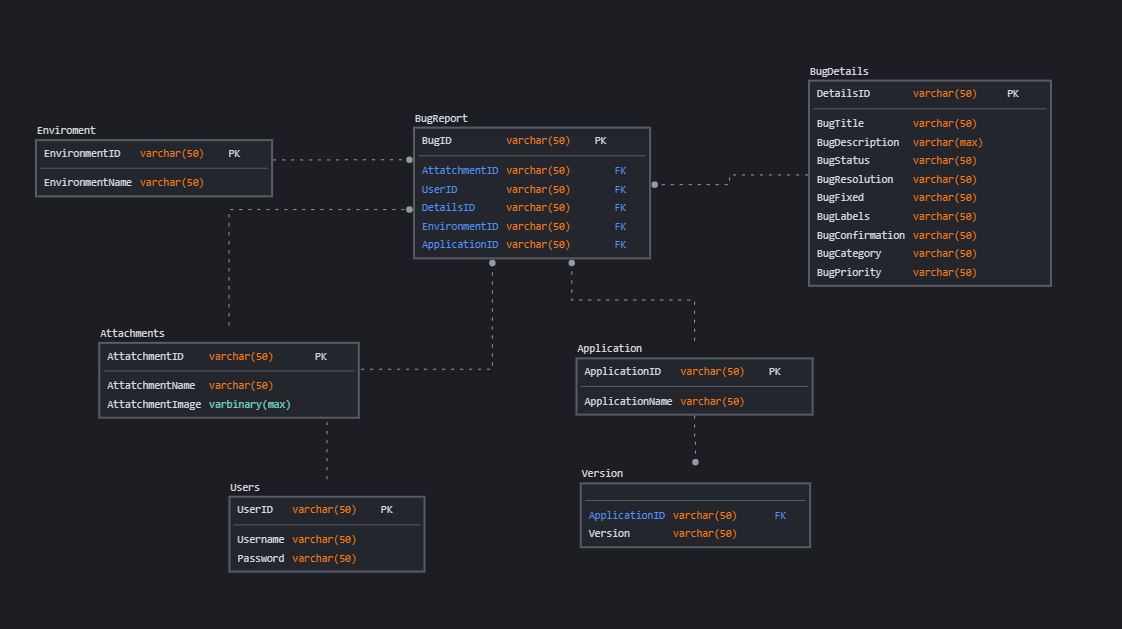
Bug images:

* Unique identifier for the bug and the image
* Images of the bug working

User information:

* Username and password
* Comments created
* Bugs solved
* Bugs reported

Entity relationship model



Users

This system will support three types of user privileges, Customer, Employee, and general user. Customers will have access to all functions, and the employees will have access to most functions with the general user having the least amount of access. The general user should be able to do the following functions:

* Create a bug report
  + Upload an image about the bug
  + Describe how the bug occurred
  + Give details about the bug
* Delete bug reports
* Create new applications and versions

Employees will have the above functions with some additional ones below:

* Create comments
* Login system

The customer will have the most functions with these additional ones:

* Create a report with high importance
* Generate reports of performance and number of bugs within the system

Operating environment

The operational environment will include:

* Platform: C#/SQL
* Database: Relational SQL database
* Operating system: Windows.

Design constraints

The program will be run in C# as a desktop application which means a dedicated server is required to run queries for the application and to allow for communication between applications.

If two users are editing the same log who gets priority or will both be saved separately without overriding them and losing data.

System features

The bug tracker system maintains information about bug reports, images of bugs, comments, and users. The priority of the project is low as development teams have been working without this software for decades, however the implementation of the program into a working environment should provide useful information and serve as a good communication tool.

Use case

* Create a new bug log
* Input name description and other information
* View created report and edit existing report
* Change description and add an image

Database

The database system implies that a single application should be able to create and manage its own database as well as connecting to a central database located within the network.

Due to the two database designs there needs to be some software to determine which server to connect to and if there is no current server to create one.

The server will act as the communication network between all connected devices, this is simpler than having peer-to-peer or peer to server connections. This can be done within this program as there is no sensitive information or direct messages which need to be sent on a peer-to-peer basis, the database only stores comments which are not private and do not require any security.

The server will store encrypted passwords; however, this may not even be required as the company will have control of the server in most cases as its less of a security risk than having a 3rd party controlling the server. Due to this case passwords may not need to be encrypted but it is advised as bad actors can still mess with the system in the case these passwords leak.

External interfaces

The user will be required to run the application on windows or a pc that supports C# applications, the product will be developed and tested using a windows environment.

The front-end software will be a C# based application with the back-end being controlled by SQL

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | Windows was chosen as it is the environment which all the programming and testing is occurring on and its on of the most widely used operating systems. |
| Database | To save the Bug reports we have chosen an SQL database. |
| C# | To implement the project, we have chosen C# for its usefulness in implementing a UI and interacting with an SQL server |

This project has low system requirements to run as no calculations and little computation is used for the functioning of the program. This should allow for a wide variety of system specifications being able to run the software. But the target audience are software developers who should have computer specifications which far exceed the requirement to run the program.