

# Bugtracker project spec/ plans

## Contents

- Bugtracker project spec/ plans ..... 1
- Bug tracker spec..... 2
- User stories ..... 2
- Database layout ..... 3
  - Bugs:..... 3
  - User: ..... 3
  - Posts:..... 3
  - Follow:..... 3
- Previous Bugtracker ..... 4
- Plans ..... 7

## Bug tracker spec

The plan is to make an app which can log bugs and allow a user to post steps taken to fix them.

I have made an early version of this as a website but I want this to use windows forms.

The user stories assume many users can interact with the app, which I believe would be more impressive.

We are not using local .mdf files as we are using a cloud-hosted DB called ClearDb. This is being hosted on Heroku and I will walk you through the steps required to connect to this.

## User stories

1. I want users to be able to register and log in, so a user can track bug's they've logged/ worked on/ fixed
2. A user should be able to store details of a bug they have found, such as a title, a description, a priority etc
3. Users should be able to see a list of all logged bugs
4. Users should be able to store updates on the steps they have taken to fix a bug
5. Bugs should be able to have their status updated once they are fixed
6. A user should be able to communicate with a fix posted on their bug (in case the fix doesn't work)
7. Bugs should be organised into 'projects' to keep them grouped together (?)
8. A user should be able to take 'ownership' of a bug to prevent multiple people from working on it (? Can do away with 'solver' idea and allow multiple users to post fixes if they want)
9. Users contributions should be tracked (how many fixes posted etc)

## Database layout

### Bugs:

- Name
- Description
- Priority
- Screenshot?
- Location
- Status (unclaimed, claimed, solved, fix posted)
- timestamp

### User:

- Username
- Password
- bugs posted / fixed

### Posts:

- Bug ID
- Text body
- userID
- timestamp
- 

### Follow:

Will allow a user to follow a bug

BugID  
userID

I believe this will do for a basics of the DB, we can add things as needed.

## Previous Bugtracker

As mentioned, I have made a version of this as a webapp, however it is only hosted locally. I pretty much finished it and it only needed some UI work to finish but I hate UI and it makes me tired just thinking about doing it

### Homepage

**BugTrack** [new](#) [Solved Bugs](#) [Notifications](#) [Log Out](#)

Found a bug? click the button, give the bug a title, a priority, tell us where the bug has been found and give it a description with this form

**Report a Bug**

Title (short description of bug):

Priority:

Location:

**Report**

ID	TITLE	LOCATION	PRIORITY	STATUS
28	Pressing bugs shwing up on claimable bug list	HomePage	low	unclaimed
39	unclaiming a bug deletes all updates on that bug	bugpage when unclaiming a bug	med	unclaimed
43	logged out users can posy bugs	as title	high	unclaimed

After a user registered and logged in (features that come with the Django framework this is based on) they would see this screen. A user is able to enter details about a bug and post them.

A user can also view bugs that have not been claimed

I tried to implememnt some kind of ownership of bugs such that a user could become a bug's 'solver' However I do not intend to keep this feature for bugtrackerv2 as it's complex and sites like stackoverflow do not use this ideology. We will discuss it.

Title (short description of bug):

Priority:

Location:

**Report**

**Bug made Successfully!**  
☐ Prevent this page from creating additional dialogues  
**OK**

Reporting a bug gets you a success message

ID	TITLE	LOCATION	PRIORITY	STATUS
28	Processing bugs showing up on claimable bug list	HomePage	low	unclaimed
39	unclaiming a bug deletes all updates on that bug	bugpage when unclaiming a bug	med	unclaimed
43	logged out users can post bugs	as title	high	unclaimed
44	a title	where bug was found	low	unclaimed

Bug now shows in the list

BugTrack [new](#) [Solved Bugs](#) [Notifications](#) [Log Out](#)

28 Processing bugs showing up on claimable bug list

Posted by: [3 new](#) Bug's Solver: None, become the bug's solver with the button above Current status: **unclaimed**

Lots like bugs that are 'processing' are showing up on the list of claimable bugs on the homepage when only unclaimed bugs should appear there. Bugs 11, 22 and 23 are all 'claimed' but are still displaying and bug 25 is processing but still visible

Solve this bug!

Clicking on a bug gets you the details and (if bug is unclaimed) the option to claim it

BugTrack [new](#) [Solved Bugs](#) [Notifications](#) [Log Out](#)

44 a title

Posted by: [3 new](#) Bug's Solver: None, become the bug's solver with the button above Current status: **claimed**

aa

Post an update on this bug:

processing

Submit Query

If you claim a bug then you get the option to post updates on it, also giving you the option of setting a new status for the bug

44 a title

Posted by: [3 new](#) Bug's Solver: [new](#) Current status: **solved**

aa

A solution to this bug has been posted! Please read the solution and try it, if it works click the 'Bug solved!' button

This solution worked! This solution didn't work!

Dec. 16, 2020, 2:32 p.m., New solver : new

Dec. 16, 2020, 2:33 p.m., Turn it off and on again

As the update's status was 'solved' The poster has the option of selecting if the 'solution' worked or not

## 44 a title

Posted by: 3 [new](#) Bug's Solver: [new](#) Current status: **solved**

aa

A solution to this bug has been posted! Please read the solution and try it, if it works click the 'Bug solved!' button

This solution worked!

This solution didnt work!

Post an update on this bug:

Turning it off and on again didnt fix it

processing

Submit Query

Dec. 16, 2020, 2:32 p.m. , New solver : new

Dec. 16, 2020, 2:33 p.m. , Turn it off and on again

Selecting 'no' prompts a user to advise why the solution didn't work and allows the user to change the bug's status

This used to display a box underneath the 'solved' update, however I must have broken this functionality lmao

The idea was to allow some kind of communication between the bug's solver and the user who posted the bug, which is fairly important.

Clicking 'yes the solution worked' sets the bug's status to solved and takes it out of the list of bugs to be solved on the front page.

Users also have profile pages where they can see details of bugs they have posted / bugs they have solved etc

**BugTrack** **new** Solved Bugs Notifications Log Out

## new's profile page

Some info on user, problems posted, solved, current bugs solving / found

No bugs solved yet, 36 bugs posted

Currently Solving Bugs Solved Posted Bugs Bugs Awaiting Solver

bugs this user solved:

6 Testing bug lifecycle Test low finished solved by new

7 newSolver element of bug is not JSON serialisable 145 in views.py high finished solved by new

8 Logic error in how solvers are found 156 views.py high finished solved by new

10 If bug isn't claimed. can't show the bugPage line 49 in models, 113 in views.py, when trying to open bugpage high finished solved by new

9 Update function doesn't know what Bug an update is being posted on 156 views.py high finished solved by new

12 test here low finished solved by new

13 When bug is finished, user can see unintended divs bugpage html low finished solved by new

15 Submit update button not POSTing bugPage med finished solved by new

## Plans

So the plan is to essentially simplify and remake the bugtracker as a windows forms application. If we can implement the functionality from the old app and make it look good then I think it would be a good project to have on your portfolio.