**Faculdade de Engenharia da Universidade do Porto**

****

**CityFix - Product Backlog**

Laboratório de Desenvolvimento de Software

Cristóvão Silva

Georgina Esteves

Márcio Fontes

Ricardo Castro

Tiago Miranda

October, 2015

**Introduction**

This artifact contains the specification of all actors that interact with the system and their respective *user* *stories* as an agile documentation of the project requirements. This section pretends to illustrate the communication between actors (with *case diagrams*) and describe the potential interactions from users with the system – *user stories*.

**1. Product Backlog and Product Backlog Items**

**2. Actors**

An actor is a person, organization, or external system that plays a role in one or more interactions with a system. Actors may inherit from another actors and are never part of the system that is being modeled (Ambler, 2004). An action performed by an actor implies a response from the system. For **CityFix** system let’s consider the following actors represented in Figure 1 (Annex 1) and described in Table 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Identifier** | **Description** | **Example** |
| 0 | Visitor | Unauthenticated user; may register or log into the system and access to public information (homepage, list of tickets) | n/a |
| 1 | Authenticated | Authenticated user; may logout from the system, edit his profile or recover his password | tlucio |
| 2 | User | Can upload and vote on a ticket | joesteves |
| 3 | Moderator | Responsible for ticketing management (validate its content, assign extra points to users based on the points system, etc.) | cris92 |
| 4 | Entity | Responsible for retrieving information from the applied tickets and updating the current tickets’ state. | cmp |
| 5 | Administrator | Responsible for users management, system security and data integrity. | admin |
| 6 | Google API | External API which will be used to detect the user’s geographical localization. | gmaps |

**3. User Stories**

An user story is a high-level definition of a requirement, containing all the enough information to make it possible to produce a reasonable estimate of the effort to implement it (Ambler, 2004). The description of a potential interaction from an user with the system focuses the behaviour requirements, rather than design aspects. Usually, an user story is described by the following template (Cohn, 2004):



For **CityFix** system, let’s consider the following *user* *stories* described on the tables below, based on the above template.

**3.1 Visitor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Name** | **Priority** | **Description** | **Cost** |
| US001 | Login | High | As a Visitor I want to login into the system so that I can have access to restricted information. | 3 |
| US002 | Register | High | As a Visitor I want to register myself so that I can upload new tickets. | 5 |
| US003 | Password Recovery | Medium | As a Visitor I want to recover my authentication credentials so that I can access the platform if I forget my password or username. | 3 |
| US004 | List Tickets | Medium | As a Visitor, I want to see a list with a simple view of recent tickets so that I know the most recent activity. | 2 |
| US005 | Facebook Register | Low | As a Visitor, I want to register myself with my Facebook account so that my basic information is already filled automatically . | 8 |

**3.2 Authenticated**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Name** | **Priority** | **Description** | **Cost** |
| US101 | Search | Medium | As an User I want to search all public information (entity profiles, etc.) so that I can know who I should contact if there is a problem near my location. | 5 |
| US102 | View Ticket | High | As an User I want to view a ticket’s content so that I can know if there is a problem near my location and what actually happened. | 3 |
| US103 | Logout | High | As an Authenticated I want to be able to logout from the system so that I can terminate my session correctly. | 1 |
| US104 | Profile | Medium | As an Authenticated I want to see and edit my profile so that I can change my basic information and authentication credentials. | 3 |

**3.3 User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Name** | **Priority** | **Description** | **Cost** |
| US201 | Add Ticket | High | As an User I want to submit a ticket to the system so that I can show that there’s a problem near my location. | 3 |
| US202 | Edit Ticket | High | As an User I want to edit a ticket i own in the system so that I can update its information. | 5 |
| US203 | Remove Ticket | High | As an User I want to remove a ticket I own so that I can hide its content from the public. | 2 |
| US204 | Associate Category | High | As an User I want to associate a ticket to a category so that it helps to solve a problem. | 1 |
| US205 | Manual Location | Medium | As an User I want to manually add a location so that I can add a ticket after a situation when I didn’t have internet connection. | 8 |
| US206 | Automatic Location | High | As an User I want the system to automatically detect my location so that I don’t have to add it manually. | 8 |
| US207 | Vote | High | As an User I want to vote on a ticket so that I can contribute to the truthfulness of that ticket and its respective user. | 3 |
| US208 | List My Tickets | Medium | As an User I want to have access to a list of my submitted tickets so that I can keep track of their status. | 3 |
| US209 | Associate Image to Ticket | High | As an User I want to associate an image to a ticket so that an user or entity can easily detect what’s happening on that location. | 5 |
| US210 | Ticket Suggestion | Medium | As an User I want to suggest changes to a ticket so that whenever I think an update must be, it shall be considered. | 5 |
| US211 | Ticket Search | High | As an User I want to search all public tickets so that I can see information on tickets others submitted | 3 |

**3.4 Moderator**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Name** | **Priority** | **Description** | **Cost** |
| US301 | Validate Tickets | High | As a Moderator I want to validate tickets so that an entity can solve the problem. | 1 |
| US302 | Change Ticket Information | Medium | As a Moderator I want to change information of a ticket so that I can update the current status, etc. | 5 |

**3.5 Entity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Name** | **Priority** | **Description** | **Cost** |
| US401 | Entity tickets | High | As an Entity I want to access the tickets associated with me so that I can take action on solving those problems. | 3 |
| US402 | Change Ticket State | High | As an Entity I want to change a ticket’s state so that an active user may know the current state of that specific problem. | 1 |
| US403 | Filter Tickets | Low | As an Entity i want to filter tickets so that I can optimize my schedule. | 8 |

**3.6 Administrator**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | **Name** | **Priority** | **Description** | **Cost** |
| US501 | Manage Users | High | As an Administrator I want to manage users so that I can delete or mute an user if necessary. | 3 |
| US502 | Manage Content | High | As an Administrator I want to manage the system’s content so that whenever a change is necessary, there is permission to do so. | 5 |
| US503 | Statistics | Low | As an Administrator I want to get statistics so that I can know information about my system. | 8 |

**3.7 Business Rules**

|  |  |  |
| --- | --- | --- |
| **Identifier** | **Name** | **Description** |
| BR01 | Password | On the registration, password must be at least 6 characters and contain numbers and letters |
| BR02 | Email | An email must be associated with only one account |
| BR03 | Ban | An user is banned whenever he uses the system improperly |
| BR04 | Aprove | Only a moderator or an  administrator can aprove a ticket |
| BR05 | Change Information | After being aproved, a ticket can not be changed |
| BR06 | Access | After authentication, the user must terminate the session before he can login with another account |
| BR07 | Ticket Text | A ticket must have between 1 and 5000 characters |
| BR08 | Ticket Creation | After creating a ticket a user has 30 minutes to change any information on it |
| BR09 | Facebook Restrition | An entity or Admin cannot have a facebook account associated |
| BR10 | Service Association | An entity can only take action on tickets whose category the service serves |

**3.8 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Identifier** | **Name** | **Description** |
| FR01 | Usability | The system must be simple and easy to use. |
| FR02 | Accessibility | The system must be available to all web browsers. |
| FR03 | Portability | The system must work on different operative systems. |
| FR04 | Performance | The system must have response time of at most 5 seconds. |
| FR05 | Security | The system must protect information from unauthorized access through a system of authentication and privilege checking. |
| FR06 | Fault tolerance | The system must be prepared to treat and keep working with runtime errors. |
| FR07 | Traceability | The system must store action history of its users, such as ticket uploads. |
| FR08 | Ethics | The system must respect the ethical principles by encrypting passwords. |
| FR09 | Scalability | The system must have the capacity to maintain the same performance when there is an increase in the number of users and / or simultaneous requests. |
| FR10 | Availability | The system must be avilable 99% of the time for each 24 hours period. |

**4. Acceptance Tests**

**4.1 Visitor**

**4.1.1 User Login**

**ID**: 1

**Requirement ID**: US001

**Description**: This test should verify that a user is able to authenticate himself in the system and take full use of its API according to his privileges.

**Actors**: Visitor

**Pre-conditions**:

* User device has active internet connection
* No account in local device authenticated in the system
* Record of account with email user@test.com with password User123 must be registered in the database.
* Record of account with email phony@test.com must not be registered in the database.
* Record of account with email moderator@test.com with password Moderator123 must be registered in the database.
* Record of account with email entity@test.com with password Entity123 must be registered in the database.
* Record of account with email admin@test.com with password Admin123 must be registered in the database.
* Record of account with account registered with facebook must be registered in the database
* No facebook session should be live
* 3.0.2 Visitor Register
* 3.0.5 Facebook Register

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access City Fix welcome page and press register login | A register form appears |
| 2 - User authentication scenario | | |
| 2.1 | Wrong email and password scenario.  Fill email login input with [phony@test.com](mailto:phony@test.com) and Password input with a password that validates system requirements (e.g.: Phony12345).  Click Login. | A message with text “wrong email/password combination” and a link with text “Recover Password” are displayed. |
| 2.2 | Wrong password scenario. Fill email login input with [user@test.com](mailto:user@test.com) and Password input with User12345. Click Login. | A message with text “wrong email/password combination” and a link with text “Recover Password” are displayed. |
| 2.3 | Valid login scenario  Fill email login input with [user@test.com](mailto:user@test.com) and Password input with User123 Click Login. | User’s homepage is loaded. User’s first name and last name are displayed on top bar. |
| 3 – Moderator authentication scenario | | |
| 3.1 | Fill email login input with [user@test.com](mailto:user@test.com) and Password input with User123 Click Login. | Login succeeds.  Moderator’s homepage is loaded. User’s first name and last name are displayed on top bar. Dropdown list on top right corner has a “moderator” item. |
| 4 – Administrator authentication scenario | | |
| 4.1 | Fill email login input with [admin@test.com](mailto:admin@test.com) and Password input with Admin123 Click Login. | Login succeeds.  Administrator’s homepage is loaded. Admin panel loads on page. |
| 5 – Entity authentication scenario | | |
| 5.1 | Fill email login input with [entity@test.com](mailto:entity@test.com) and Password input with Entity123 Click Login. | Login succeeds.  Entity’s homepage is loaded. Entity’s panel loads on page.  Displays ticket’s to be served by the entity |
| 6 – Facebook User authentication with no facebook session live scenario | | |
| 6.1 | Press login with facebook | A popup asking to provide facebook login information (if a facebook session is live) shows up. |
| 6.2 | Fill facebook form with valid facebook account also registered in CityFix. Press login. | User’s homepage is loaded. User’s first name and last name are displayed on top bar. |
| 7 – Facebook User authentication with live facebook session scenario | | |
| 7.1 | Login to facebook | Facebook interface will appear |
| 7.2 | Open CityFix welcome page again | CityFix page will appear |
| 7.3 | Press login with facebook | User’s homepage is loaded. User’s first nameand last name are displayed on top bar. |

**4.1.2 Visitor Register**

**ID**: 2

**Requirement ID**: US002, BR01, BR02

**Description**: This test should verify that a visitor is able to register an account in the system and take full use of its API according to his privileges.

**Actors**: Visitor

**Pre-conditions**:

* User device has active internet connection
* No account in local device authenticated in the system
* Record of account with email [user@test.com](mailto:user@test.com) in the database

**Post-conditions**:

* Repeat test 3.0.1 – User Login through step 2.3 with email account used in registration step 5.

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access City Fix welcome page and press register | A register form will appear |
| 2 | Mandatory fields missing scenario.Leave all fields blank and select register | An alert message with text “Some fields are missing” is displayed. Missing fields are also highlighted. |
| 3 | Unique email violation scenario.  Fill all fields with correct criteria except email. Insert [user@test.com](mailto:user@test.com) on email input and Click Register | An alert message with text “Email account already registered” is displayed. Missing fields are also highlighted. |
| 4 | Password requirements violation scenario. Fill all fields with correct criteria except password. Insert “111” in password field and press register | Message saying “Password must be |
| 5 | Correct registration scenario. Fill all fields with correct criteria (email cannot be [user@test.com](mailto:user@test.com)). An email account to which you have access must be provided. Press register | A message displaying “Welcome to City fix. An email was sent to you. Please confirm your registration”. |
| 6 | Repeat test 3.0.1 - User Login through step 2.3 with email account used in registration step 5. | Login fails due to wrong email and password.  A message with text “wrong email/password combination” and a link with text “Recover Password” are displayed. |
| 7 | Go to email account used in registration in step 5, open confirmation email and click the provided link. | Page with message “Registration confirmed” will load. |

**4.1.3 Visitor Password Recovery**

**ID**: 3

**Requirement ID**: US003, BR01

**Description**: This test should verify that a visitor is able to recover his password.

**Actors**: Visitor

**Pre-conditions**:

* User device has active internet connection
* No account logged in on device
* Record of account with email [phony@test.com](mailto:phony@test.com)  cannot be in the database.
* Test 3.0.2 Visitor Register until step 7

**Post-conditions**:

* Repeat test 3.0.1 – User Login through step 2.3 with email account used in password recovery step 7.

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access City Fix welcome page |  |
| 2 | Click recover password | A form will appear with a password input |
| 3 | Non-registered account scenario.  Fill data with [phony@test.com](mailto:phony@test.com) and press recover | A message with text “the email account provided is not registered” will be displayed |
| 4 | Valid account scenario.  Fill data with valid email (email registered in and press recover | Message saying to check email will appear |
| 5 | Go to email account and click recover password | A page will load with input for new password |
| 6 | Password requirements violation scenario.  Insert password “123”. | A message will load saying password does not meet criteria |
| 7 | Fill password input with a password that validates system requirements (e.g.: Phony12345). | Messsage with text “A new password has been set please login” will be displayed. |

**4.1.4 List Tickets**

**ID**: 4

**Requirement** **ID**:

**Description**: This test should verify that a list of recent tickets is shown on the welcome page

**Actors**: Visitor

**Pre-conditions**:

* User device has active internet connection
* Record of at least one ticket in database

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access City Fix welcome page | Information about CityFix, |
| 2 | Scroll down until a list with label “Recent tickets” show up | A list with one or more recent tickets will appear |

**4.1.5 Facebook Register**

**ID**: 5

**Requirement ID**: US005

**Description**: This test should verify that a visitor is able to register himself in the system using facebook authentication.

**Actors**: Visitor

**Pre-conditions**:

* User device has active internet connection
* User must have facebook account.

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access City Fix welcome page |  |
| 2 | Press register with facebook | A popup asking to provide facebook login information or to confirm facebook account details (if a facebook session is live) shows up. |
| 3 | Click accept | A message with text “Welcome to City fix.” appears |

**4.2 Authenticated**

**4.2.1 Search**

**ID**: 6

**Requirement ID**: US101

**Description**: This test should show to an authenticated user the result of his/her search.

**Actors**: Authenticated

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1. | Locate the search bar |  |
| 2. | The authenticated user should write what he/she wants to search |  |
| 3. | The authenticated user should click on the “Ok” button | A page with his/her search results will appear |

**4.2.2 View Ticket**

**ID**: 7

**Requirement ID**: US102

**Description**: This test should show to an authenticated user the details of a specific ticket.

**Actors**: Authenticated

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;

|  |  |  |
| --- | --- | --- |
| Step | Description | Results |
| 1. | Click on “Tickets” button | A list of tickets appears |
| 2. | Select one of the tickets | The detailed information of that ticket is presented. |

**4.2.3 Logout**

**ID**: 8

**Requirement ID**: US103

**Description**: This test should make an authenticated user go unauthenticated.

**Actors**: Authenticated

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;

|  |  |  |
| --- | --- | --- |
| Step | Description | Results |
| 1. | Click on “Logout” button | The authenticated user is logged out. |

**4.2.4 Profile**

**ID**: 9

**Requirement ID**: US104

**Description**: This test should show to an authenticated user his/her own profile page where he/she can see and edit his/her own information

**Actors**: Authenticated

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;

|  |  |  |
| --- | --- | --- |
| Step | Description | Results |
| 1. | Click on “My Profile” button | The profile page is displayed |
| 2. | Click on “Edit Profile” button | The user can now edit his/her basic information |

**4.3 User**

**4.3.1 Add Ticket**

**ID**: 10

**Requirement ID**: US201

**Description**: This test should show to an user a submitted ticket by him/her

**Actors**: Authenticated

**Pre-conditions:**

* User device has active internet connection;
* The user must be logged in;

|  |  |  |
| --- | --- | --- |
| Step | Description | Results |
| 1. | Click on “Add Ticket” button | A page to submit a ticket is displayed |
| 2.1 | All informations about the ticket are valid | Ticket is submitted |
| 2.2 | Informations about the ticket are not valid | Ticket not submitted |
| 3 | User saves the changes | Data verification is made |
| 4.1 | The changes are successful | The data is updated |
| 4.2 | A problem occured while changing the data | The data remains the same and the user returns to the previous page |

**4.2.2 Edit Ticket**

**ID**: US202

**Requirement ID**: US202

**Description**: This test should verify if a user can ticket edit its, i.e, change its current status, respective description, etc.

**Actors:** User

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;
* The ticket must belong to the respective user;
* The ticket must not be blocked by the moderator (concerning the status) in order to edit it;

**Post-conditions**:

* The ticket’s respective status must not change if it is already blocked by a moderator;
* If no problem occurs, then the changes must be saved and the data consistent;
* Otherwise, the data must remain unchanged;
* The user redirected to his Tickets page;

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “My Tickets” button | A list of own tickets appears |
| 2.1 | Select one of the tickets | The detailed information of that ticket is displayed |
| 2.2 | There are no tickets to be displayed | The user can’t edit any tickets |
| 3 | A list of fields is displayed and the user must edit the respective data |  |
| 4 | User saves the changes | Data verification is made |
| 5.1 | The changes are successful | The data is updated |
| 5.2 | A problem occured while changing the data | The data remains the same and the user returns to the Tickets page |

**4.3.3 Remove Ticket**

**ID**: US203

**Requirement ID**: US203

**Description**: This test should remove a ticket from an user (if is not blocked or undergoing).

**Actors:** User

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;
* The ticket must belong to the respective user;
* The ticket must not be blocked by the moderator (concerning the status) in order to edit it;

**Post-conditions**:

* If no problem occurs, then the ticket is removed;
* Otherwise, the ticket must remain unchanged;
* The user redirected to his Tickets page;

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “My Tickets” button | A list of own tickets appears |
| 2.1 | Select one of the tickets | The detailed information of that ticket is displayed |
| 2.2 | There are no tickets to be displayed | The user can’t edit any tickets |
| 3.1 | Click on the delete button |  |
| 3.2 | The delete button is blocked due to ticket’s undergoing status | The user can’t remove the ticket |
| 4 | An input box appears and asks the user if he really wants to remove the ticket | The user must choose yes or no |
| 5.1 | The changes are successful | The ticket is removed |
| 5.2 | A problem occured while removing the ticket | The ticket remains in the database |

**4.3.4 Associate Category**

**ID**: US204

**Requirement ID**: US204

**Description**: This test should test if it allows an user to associate a category to a ticket.

**Actors:** User

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;
* The ticket must belong to the respective user (if the user wants to associate a category after creating it);
* The ticket must not be blocked by the moderator (concerning the status) in order to edit the category;

**Post-conditions**:

* If no problem occurs, then the category is associated;
* Otherwise, the ticket must remain unchanged or not created;
* The user is redirected to his Tickets page;

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “My Tickets” button | A list of own tickets appears |
| 2.1 | Click on “Create a Ticket” button | A creation form appears |
| 2.2 | Select one of the tickets | The detailed information of that ticket is displayed |
| 3.1 | User fills the form and then selects a category |  |
| 3.2 | A list of fields is displayed and the user must edit the category |  |
| 4 | User clicks on the “Save” button | Data verification is made |
| 5.1 | The changes are successful | The ticket is created or edited and a category is associated |
| 5.2 | A problem occured while creating or editing the ticket and associating a category | The category is not associated |

**4.3.5 Manual Location**

**ID**: US205

**Requirement ID**: US205

**Description**: This test should test if it allows an user to type a location during a ticket creation.

**Actors:** User

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;

**Post-conditions**:

* If no problem occurs, then the ticket is created with the specified location;
* Otherwise, the ticket is not created and the user must specify the location again;
* The user redirected to his Tickets page;

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “My Tickets” button | A list of own tickets appears |
| 2.1 | Click on “Create a Ticket” button | A creation form appears |
| 3 | User fills the form and then types the location | The system tries to establish connection to Google Maps API to recognize the typed location for a more efficient use. |
| 4.1 | The typed location is not recognized by Google Maps | The location is still saved but not directly connected to Google Maps |
| 4.2 | The typed location is recognized by Google Maps | The location stays associated to Google Maps |
| 5 | User clicks on the “Save” button | Data verification is made |
| 6.1 | The changes are successful | The ticket is created and the location is associated |
| 6.2 | A problem occured while creating the ticket or associating a location | The ticket is not created and the user must rewrite or try again to recognize the location |

**4.3.6 Automatic Location**

**ID**: US206

**Requirement ID**: US206

**Description**: This test should test if it allows an user recognize its current location.

**Actors:** User

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;
* The device must be capable to recognize the user’s geographical location.

**Post-conditions**:

* If no problem occurs, then the user’s location is recognized;
* Otherwise, the user must specify a manual location;

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “My Tickets” button | A list of own tickets appears |
| 2.1 | Click on “Create a Ticket” button | A creation form appears |
| 3 | User fills the form and then selects the automatic location | The system tries to establish connection to Google Maps API to recognize the user’s location. |
| 4.1 | The location is recognized | The user may save the ticket and its respective location is associated |
| 4.2 | The user attempts to recognize the location again | The system tries again to establish a connection to Google Maps API |
| 4.3 | The location is not recognized and the user must type the location by himself (4.2.5) |  |
| 5 | User clicks on the “Save” button | Data verification is made |
| 6.1 | The changes are successful | The ticket is created and the location is associated |
| 6.2 | A problem occured while creating the ticket or associating a location | The ticket is not created and the user must rewrite or try again to recognize the location |

**4.3.7 Vote**

**ID**: US207

**Requirement ID**: US207

**Description**: This test should test if it allows an user to vote on a ticket.

**Actors:** User

**Pre-conditions**:

* User device has active internet connection;
* The user must be logged in;

**Post-conditions**:

* If no problem occurs, then the user’s location is recognized;
* Otherwise, the user must specify a manual location;

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “Tickets” button | A list of tickets appears |
| 2 | Select a ticket | The detailed information of that ticket is displayed |
| 3 | The user clicks on the “Vote” button (Up or Down) |  |
| 4.1 | The vote is successful | The vote is counted and is saved on the database |
| 4.2 | A problem occured while voting (try to vote twice or lost internet connection, etc.) | The vote is saved on the database |

**4.3.8 List My Tickets**

**ID**: 17

**Requirement ID**: US208

**Description**: This test checks if the user is able to correctly list the tickets he created

**Actors**: User

**Pre-conditions**:

* The user must be authenticated with an User account
* Internet connection

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the User profile |  |
| 2 | Click on the symbol that opens the menu | The menu with multiple options will show |
| 3 | Click on “My tickets” | The list of tickets submitted by the user will show |

**4.3.9 Associate Image to Ticket**

**ID**: 18

**Requirement ID**: US209

**Description**: This test checks if the user is able to attach an image to the ticket he is creating, either by using the phone’s camera or by searching in its gallery

**Actors**: User

**Pre-conditions**:

* The user must be authenticated with an User account
* Access to the phone’s camera
* Access to the phone’s gallery
* A ticket creation process must have been initiated previously:

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “Attach a photo” | A popup asking if the user wants to take a photo or search in the phone’s gallery appears |
| 2 | Click on “Take a Photo” | A new window will open with the image from the phone’s camera |
| 2.1 | Take the photo | The photo taken will automatically appear in the ticket creation screen. The camera will close. |
| 2 | Click on “Upload from Gallery” | A new window will appear listing all the photos the user has in its phone’s gallery |
| 2.1 | Click on a photo | The photo will become selected and a button will appear in the top right corner saying “Upload” |
| 2.2 | Click on “Upload” | The selection screen will close and the photo will appear in the ticket creation screen |

**4.3.10 Ticket Suggestion**

**ID**: 19

**Requirement ID**: US210

**Description**: This test checks if the user is able to submit for approval suggestions to change some information in the ticket details

**Actors**: User

**Pre-conditions**:

* The user must be authenticated with an User account
* Internet connection

:

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Click on “Search tickets” | A new screen |
| 2 | Click on “Take a Photo” | A new window will open with the image from the phone’s camera |
| 2.1 | Take the photo | The photo taken will automatically appear in the ticket creation screen. The camera will close. |
| 2 | Click on “Upload from Gallery” | A new window will appear listing all the photos the user has in its phone’s gallery |
| 2.1 | Click on a photo | The photo will become selected and a button will appear in the top right corner saying “Upload” |

**4.3.11 Ticket Search**

**ID**: 20

**Requirement ID**: US211

**Description**: This test checks if the user is able to search the ticket directory, either by date, or location, or by entity

**Actors**: User

**Pre-conditions**:

* The user must be authenticated with an User account
* Internet connection

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the User profile |  |
| 2 | Click on the symbol that opens the menu | The menu with multiple options will show |
| 3 | Click on “Search Tickets” | A new screen will appear with text fields to fill |
| 4 | Fill the fields with required information |  |
| 5 | Click on “Search” | In the screen will appear a list of tickets corresponding to the information requested |

**4.4 Moderator**

**4.4.1 Validate Tickets**

**ID**: 21

**Requirement ID**: US301

**Description**: This test checks if the moderator is able to approve a ticket previously submitted

**Actors**: Moderator

**Pre-conditions**:

* The user must be authenticated with a Moderator account
* Access to the internet

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the user account |  |
| 2 | Search a ticket | A list of tickets corresponding to the search will appear in a new screen |
| 3 | Click on an unapproved ticket | A new window will appear will all the ticket information, as well as a button in the bottom saying “Approve ticket” |
| 4 | Click on “Approve Ticket” | The where the Moderator clicked will become green saying “Ticket Approved” |

**3.3.2 Change Ticket Information**

**ID**: 22

**Requirement ID**: US302

**Description**: This test checks if the moderator is able to change some information on the ticket’s details

**Actors**: Moderator

**Pre-conditions**:

* The user must be authenticated with a Moderator account
* Access to the internet

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the user account |  |
| 2 | Search a ticket | A list of tickets corresponding to the search will appear in a new screen |
| 3 | Click on a ticket | A new window will appear will all the ticket information, as well as “Edit” buttons on each of the ticket’s fields |
| 4 | Click on “Edit” on any field | The field will become editable and the Moderator can now change the information. A green tick mark will appear next to the field being edited. |
| 5 | Click on the green tick mark | The field will be locked and the information in the ticket is updated. The screen keeps showing ticket details |

**4.5 Entity**

**4.5.1 Entity tickets**

**ID**: 23

**Requirement ID**: US401

**Description**: This test will check that the tickets are being redirected to the right Entity, and that the Entity is able to access the tickets in order to solve the situation.

**Actors**: Entity

**Pre-conditions**:

* The user must be authenticated with an Entity account
* Internet connection

**Post-conditions**:

* The user accesses a list of active tickets associated with the Entity. Then, he can access the tickets and change the state of them, if needed.

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the Entity profile |  |
| 2 | Click on the “My Tickets” button |  |
| 3.1 | A list of active tickets is displayed, and the user can access the tickets listed by clicking it | A list of multiple active tickets is displayed, and the user can access the tickets |
| 3.2 | There are no available tickets to be displayed | A warning informing that there are not active tickets at the moment appears on the screen |

**4.5.2 Change Ticket State**

**ID**: 24

**Requirement ID**: US402

**Description**: This test will confirm that when a ticket is validated and associated with an Entity, the Entity responsible can update the current state of a situation, and even close it.

**Actors**: Entity

**Pre-conditions**:

* The user must be authenticated with an Entity account
* Internet connection
* At least one active ticket related to that Entity must be available

**Post-conditions**:

* The current state of the ticket changes

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the Entity profile |  |
| 2 | Click on the “My Tickets” button | A list of multiple active tickets is displayed |
| 3 | Select one of the tickets | The detailed information of that ticket is displayed |
| 4 | Click the dropdown menu, and choose a new state to the ticket |  |
| 5.1 | The alteration is successfully made | The state of ticket is updated |
| 5.2 | A problem occured with the selection | The state of the ticket remains the same |

**4.4.3 Filter Tickets**

**ID**:25

**Requirement ID**: 403

**Description**: This test will confirm that an Entity can successfully filter tickets by geographical distance, and with this organize an optimized route to attend and solve the active problems

**Actors**: Entity

**Pre-conditions**:

* The user must be authenticated with an Entity account
* Internet connection

**Post-conditions**:

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the Entity profile |  |
| 2 | Click on the “My Tickets” button | A list of multiple active tickets is displayed |
| 3 | Select “My route” button | A map using Google Maps is displayed, with marks representig the active problems in the area |
| 4 | Select one of the active marks | The mark is highlighted, and some information about the problem is displayed |
| 5.1 | Click the “plus” sign next to the mark icon | The mark is added to the current route |
| 5.2 | A problem occured while adding the mark | A popup appears indicating the error and the route remains the same |

**4.5 Administrator**

**4.5.1 Manage Users**

**ID**: 26

**Requirement ID**: 501

**Description**: This test will verify the user’s management system. All the activities related with change of permissions or accounts will be tested.

**Actors**: Administrator and User

**Pre-conditions**:

* The user must be authenticated with an Administrator account
* Internet connection

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the Administrator profile |  |
| 2 | Click on the “Manage Users” button | A list of the system’s users is displayed |
| 3 | Click on the name of an User, Moderator or Entity | The detailed information of that User is displayed |
| 4.1 | Click the dropdown menu, and promote an User to Moderator | The User is now seen by the system as a Moderator, and gains the permission associated with it |
| 4.2 | Click the dropdown menu, and depromote a Moderator to User | The Moderator is now seen by the system as an User, and loses his previous permissions |
| 4.3 | Click the dropdown menu, and delete an User, Moderator or Entity | The User’s account no longer exists in the system |

**4.5.2 Manage Content**

**ID**: 27

**Requirement ID**: 502

**Description:**

**Actors**: Administrator

**Pre-conditions**:

* The user must be authenticated with an Administrator account
* Internet connection

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the Administrator profile |  |
| 2 | Click on the “Manage Content” button | A list of most active tickets will be displayed |
| 3 | Click on a ticket | The detailed information of that ticket is displayed |
| 4.1 | Click the dropdown menu, and select “Edit Ticket” button | The ticket information is immediately updated |
| 4.2 | Click the dropdown menu, and select “Delete Ticket” button | The ticket is removed from the system |

**4.5.3 Statistics**

**ID**: 28

**Requirement ID**: 503

**Description**:

**Actors**: Administrator

**Pre-conditions**:

* The user must be authenticated with an Administrator account
* Internet connection

|  |  |  |
| --- | --- | --- |
| Step | Description | Result |
| 1 | Access the Administrator profile |  |
| 2 | Click on the “Manage Content” button | A view with multiple graphs and statistics is displayed |
| 3 | Click on the radio button corresponding to the desired information | The graph will change, displaying the information accordingly |

**Annexes**

**Annex A**

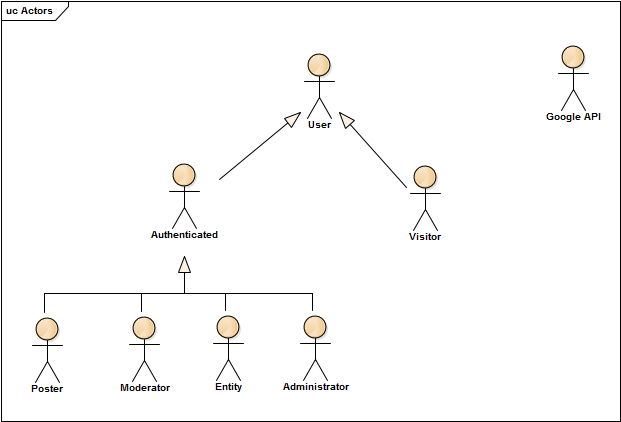


Figure 1 - Actors

**Annex B**

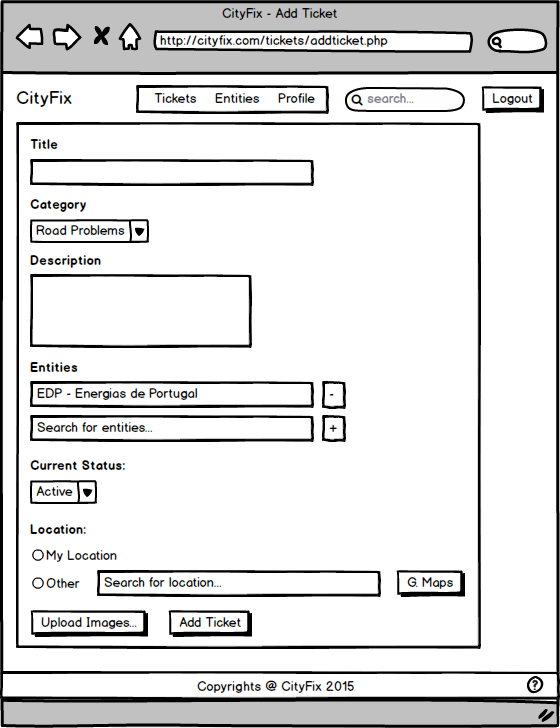


Figure 2 - Add Ticket UI

**Annex C**

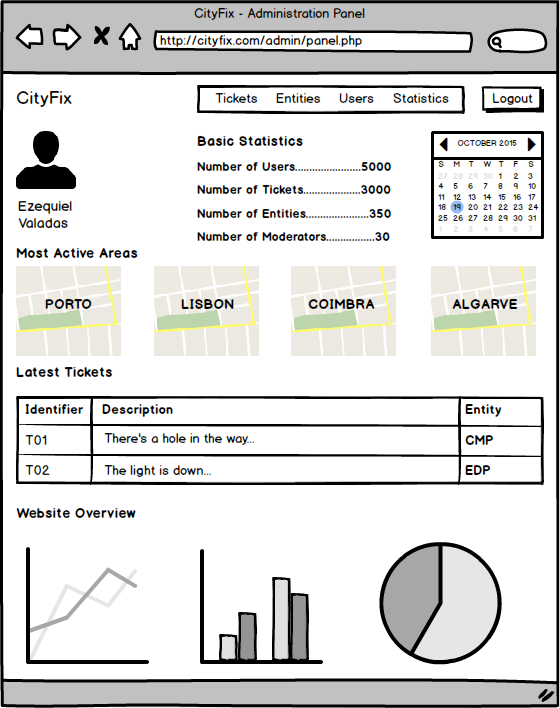


Figure 3 - Administrator Panel UI

**Annex D**

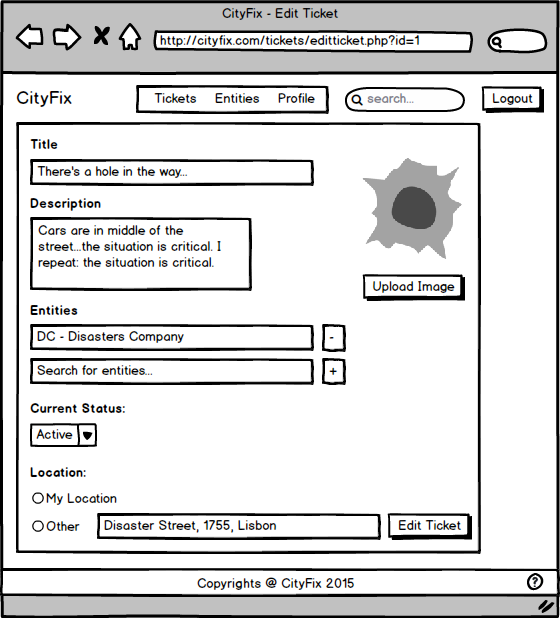
****

Figure 4 - Edit Ticket UI

**Annex E**

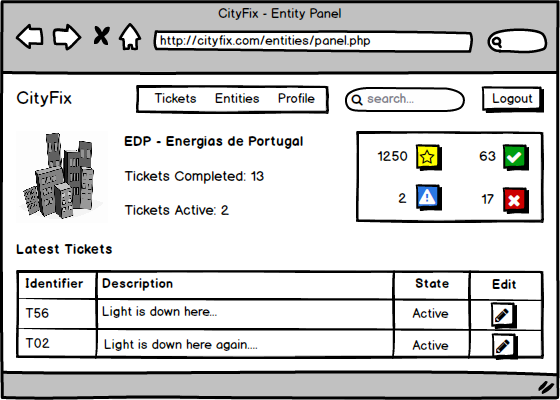
****

Figure 5 - Entity Panel UI

**Annex F**

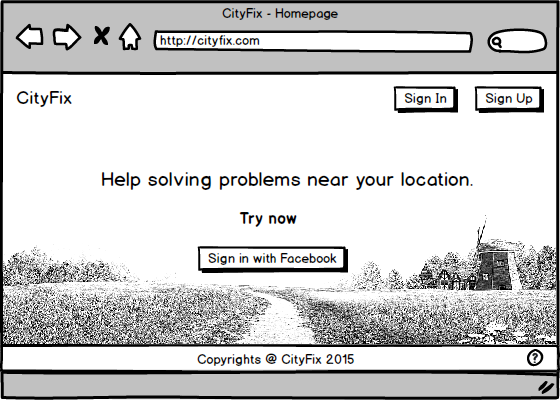
****

Figure 6 - Homepage UI

**Annex G**

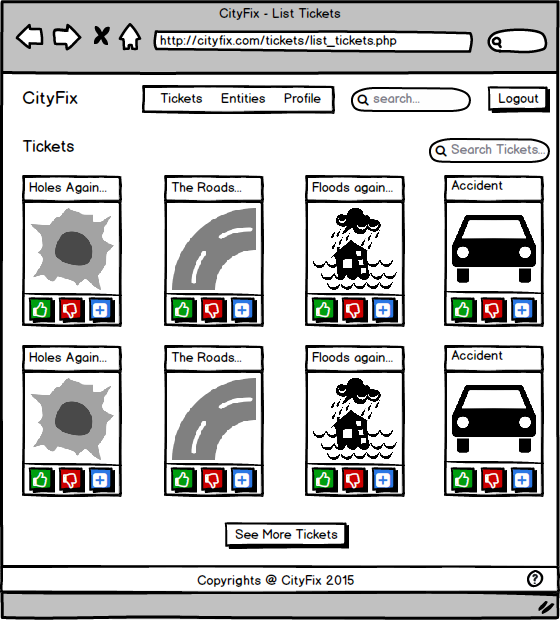
****

Figure 7 - List Tickets UI

**Annex H**

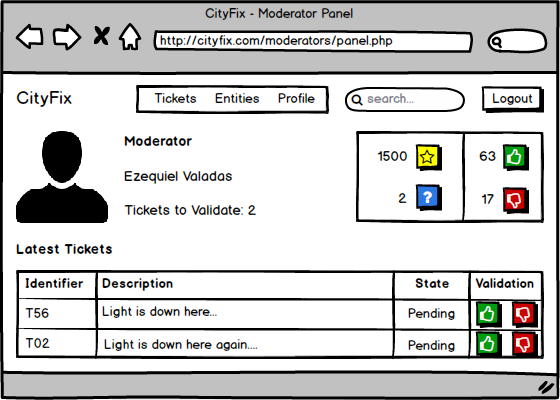


Figure 8 - Moderator Panel UI

**Annex I**

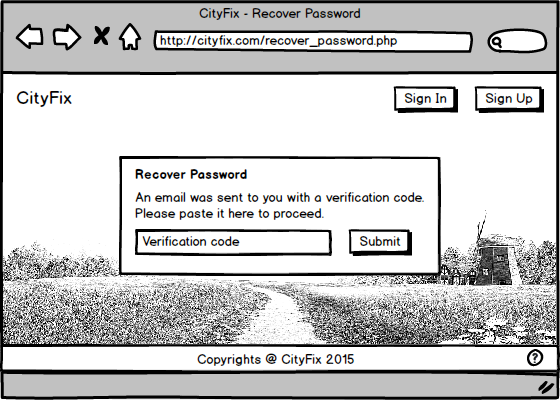


Figure 9 - Recover Password UI

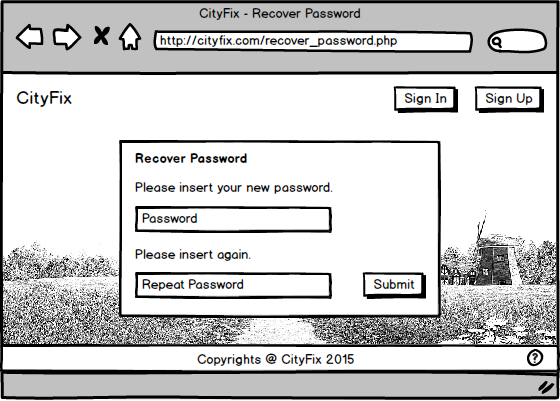
**Annex J**

Figure 10 - Recover Password 2 UI

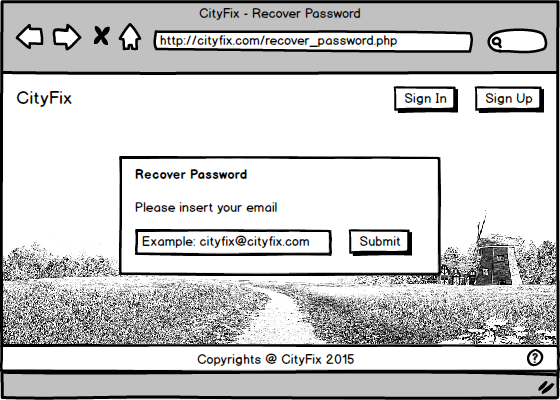
**Annex K**

Figure 11 - Recover Password 3 UI

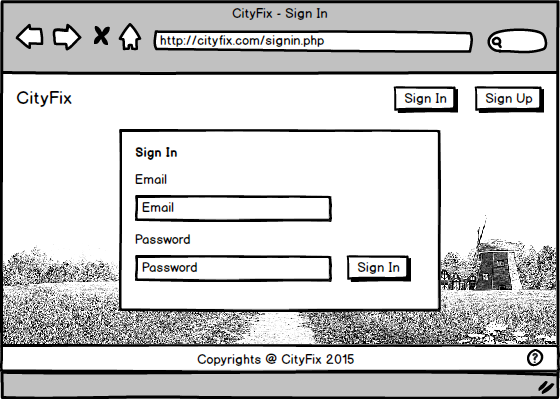
**Annex L**

Figure 12 - Sign In UI

**Annex M**

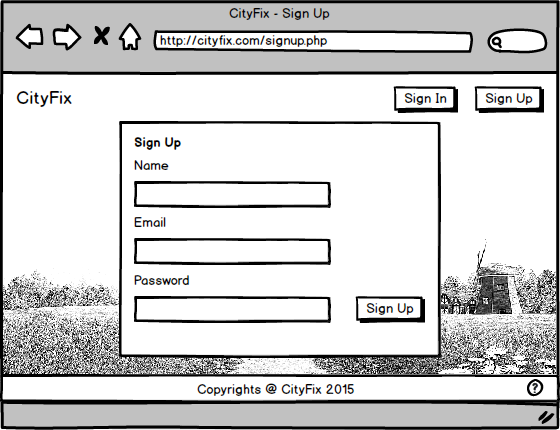
****

Figure 13 - Sign Up UI

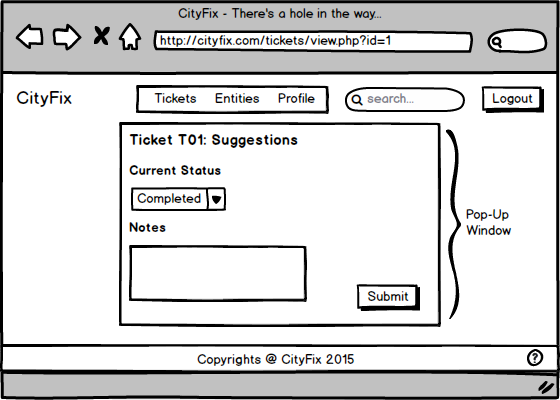
**Annex N**

Figure 14 - Suggestions to Ticket UI

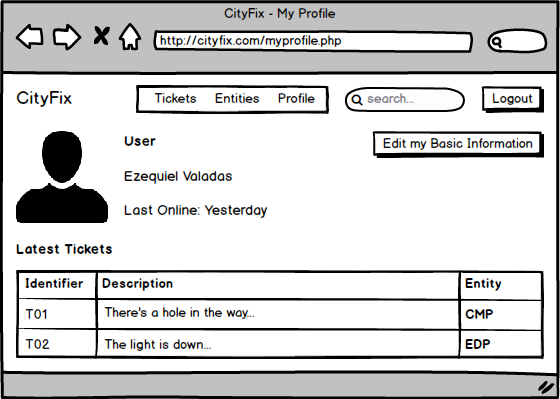
**Annex O**

Figure 15 - User Profile UI

**Annex P**

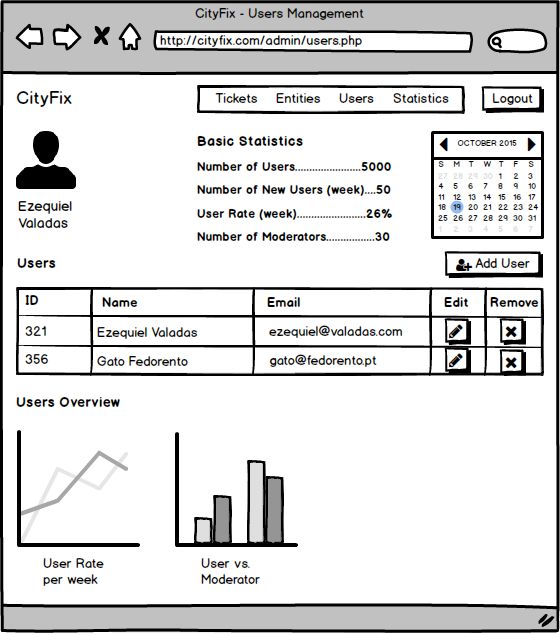


Figure 16 - Users Management UI

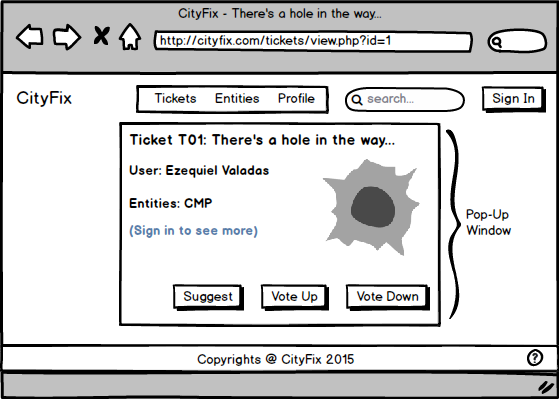
**Annex Q**

Figure 17 - View Ticket (online) UI

**Annex R**

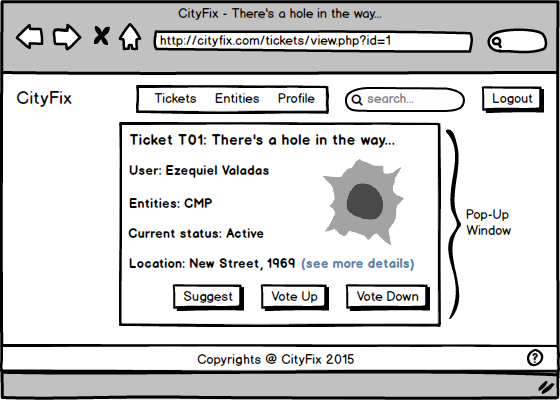


Figure 18 - View Ticket (offline) UI