**User Stories**

**Actors**

1. unregistered user
2. registered user
3. logged in user (registered and logged in user)
4. all users (registered, not registered and logged in users)
5. website owner

**Epic: Authentication**

1. **User registration (functional)**

As an unregistered user, I want to create an account, to be able to post, buy, sell and lend products on the website.

**Acceptance criteria**

* The user can create an account via join button
* A new page is displayed for the registration fields (see attachment)
* It should be clear, which fields are optional (see attachment)
* There should be a button to cancel and to confirm the registration
* There should be a check, if the registration data is valid (e.g. @ in mail address)
* There should be a check, if the user is already registered
* The user should get feedback if the registration was successful or not
* The user should be able to access the support page

**Attachment**

* List of required and optional fields for registration

**Priority: High**

**Time estimate:**

1. **Convenient registration (non-functional: usability: speed)**

As the website owner, I want that an unregistered user to be able to register fast at any time, so we don’t lose potential users.

**Acceptance criteria**

* On every page of the website which an unregistered user can access, there is a join button at the top right if the user is not logged in.

**Priority: Medium**

**Time estimate:**

1. **Secure login credentials (non-functional: safety)**

As a registered user, I want to have secure login credentials, so my account cannot be accessed by unauthorized people.

**Acceptance criteria**

* During the registration phase the user suggested password should be checked
* The password should fulfill these requirements: Combination of capital and small letters, numbers, and special characters, at least 7 characters
* If the password doesn’t meet the requirements, the user should get a message telling him about the requirements and ask him to create another password
* The password must be saved in the database with an encryption
* The user should be able to see the password if desired to make sure it is error free

**Priority: Medium**

**Time estimate:**

1. **User login (functional)**

As a registered user, I want to login or log out at any time, so I can use the functionality (just to see the buy and sell buttons for now) behind the login wall whenever I want.

**Acceptance criteria**

* The user has to enter a valid username/email and password to log in
* There should be no login without registration
* On every page on the website there is a login button, if the user is logged in, there is a logout button instead
* If the user logs in, he should have access to the functionality behind the login wall
* If the user logs out, he should not be able to use the functionality behind the login wall
* After login or logout there is a message/indication of the new status

**Priority: High**

**Time estimate:**

1. **User password reset (functional)**

As a registered user, I want to reset my forgotten password, so I can login to my account again.

**Acceptance criteria**

* During the login, the user has the option to reset the password
* To reset the password, user has to provide the username/email and answer correctly one of the random questions asked during the registration process
* If the answer is correct, he can reset the password
* If the answer is false for 3 times, the account gets blocked
* A blocked account can only be unblocked by calling a hotline and confirming their identity

**Priority: Low**

**Time estimate:**

1. **User access (non-functional: access security)**

As the website owner, I want all the users to be able to check the catalog of products but only logged in users can sell or purchase products, because the system does have necessary information to process the transaction.

**Acceptance criteria**

* All user should be able to access catalog of products
* Only logged in users should be able to post, buy, sell and lend products

**Priority: Medium**

**Time estimate:**

**Epic: User account**

1. **Transaction History (functional)**

As a registered user, I want to see which items I posted, sold, lent or purchased, so that I have a history of my actions.

**Acceptance criteria**

* There is a button called “History” for every logged in user
* When the post, sell, lend and purchase functionality gets implemented each of this action should be saved as an item in a list attached to the user
* Each item should be identifiable with the labels posted, sold, lent or purchased
* Each item should have a timestamp

**Priority: Medium**

**Time estimate:**

1. **Check the points in wallet (functional)**

As a user, I want to have a wallet, so that I can see how many points I have.

**Acceptance criteria**

* There is a button called “Wallet” for every logged in user
* Only the user has access to the wallet
* The user should not be able to do any action, that would leave him with less than 0 points

**Priority: Medium**

**Time estimate:**

1. **Default points in the wallet (functional)**

As the website owner, I want the wallets of each newly registered user to have 100 points, so they can buy products on the website.

**Acceptance criteria**

* When a new user registers, his/her wallet starts with 100 points.

**Priority: Medium**

**Time estimate:**

**Epic: System requirements**

1. **Portability (non-functional: portability)**

As the website owner, I want the website to run on Firefox, Chrome, Edge and Safari, so that I can reach a wide potential user base.

**Acceptance criteria**

* The website runs on Firefox, Chrome, Edge and Safari

**Priority: Medium**

**Time estimate:**

**Attachment**

Required fields for registration:

* Email
* Username
* Password
* First name
* Last name
* Answer to 3 question for resetting password
  + What is your mother's maiden name?
  + What is the name of your first pet?
  + What is the name of the town where you were born?

Optional fields for registration:

* Gender
* Telephone number
* Address (broken down into street, pin code, city and country)