# Product Release Requirements

## Functional Requirements (Product Backlog)

The following table shows a list of our product’s functional requirements, including for User Interface, User Manager, Proposal Manager, Group Manager, Delivery Manager, Search Engine, Other Services Manager, and Database. Functional Requirements are described in Use Cases.

* Notes:
  + Priorities are classified into rank 1 to 5, with 1-Very Low, 2-Low, 3-Average, 4-High, 5-Very High
  + All of our software’s functional requirements will be handled using Eclipse Modeling Framework (EMF).
  + Most of the use cases do not contain alternative flow of events as we want to design our product to be straightforward to use, as well as reduce the coupling/dependent degree of different functions in our software for better software quality.

Table . Use Cases List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Use Case | Actors | Priority | Includes |
| UC1 | Create Account | SaaS Operator, Customer | 5 | (n/a) |
| UC2 | Login | SaaS Operator, Customer | 5 | UC1 |
| UC3 | Search Proposal | Customer | 4 | (n/a) |
| UC4 | Make Proposal | Customer | 5 | UC2 |
| UC5 | Vote | Customer | 5 | UC4 |
| UC6 | Pay | Customer | 5 | UC10 |
| UC7 | Spread | Customer | 3 | (n/a) |
| UC8 | View Status | Customer | 4 | UC5 |
| UC9 | Evaluate Proposal | SaaS Operator | 5 | UC2 |
| UC10 | Approve Proposal | SaaS Operator | 5 | UC9 |
| UC11 | Deliver | SaaS Operator | 5 | UC6 |
| UC12 | Suggest | SaaS Operator | 3 | UC2 |

**Use Case Details:**

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC1 | |
| Use Case Name | Create Account | |
| User Story | As a user, I want to create an account so that I can access Cheaper2Get services | |
| Actors | SaaS Operator, Customer | |
| Use Case Description | Users create a new account in Cheaper2Get | |
| Pre-conditions | User Opens the home page of Cheaper2Get | |
| Post-Conditions | New account is created, login interface is ready for users  Users are added to group generated automatically | |
| Use Case Scenarios | Normal Flow | * User start Cheaper2Get software * User clicks on “Create Account” button * User fills in personal information * User Clicks on “Submit” button |
| Alternative Flows | (none) |
| Exceptions | * Software or interface cannot be loaded * ID already existed * Password unacceptable |
| Includes | (none) |
| Rules | * ID should contain at least 6 characters * Password should contain at least 8 characters including at least 1 number |
| Special Requirements | * From software to loaded interface takes less than 3 seconds |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC2 | |
| Use Case Name | Login | |
| User Story | As a user, I want to login so that I can use Cheaper2Get software | |
| Actors | SaaS Operator, Customer | |
| Use Case Description | User logs in to Cheaper2Get software | |
| Preconditions | User has created an account | |
| Post-conditions | Customer/SaaS Operator’s interface is ready for use | |
| Use Case Scenarios | Normal Flow | * User opens the software * User enters ID and Password in Login Interface * System validates the typed username and password * User successfully logs into his/her profile page |
| Alternative Flows | (none) |
| Exceptions | * Software or interface cannot be loaded * Wrong ID or password |
| Includes | UC1 |
| Rules | * Three continuous times of wrong ID or/and password in less than three minutes will prevent users from logging in the software in three minutes |
| Special Requirements | * Logging in process takes less than 10 second |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC3 | |
| Use Case Name | Search Proposal | |
| User Story | As a user, I want to Search Proposal so that I can know about the proposal(s) | |
| Actors | Customer | |
| Use Case Description | Customer searches information about suggested proposals | |
| Preconditions | User Opens the home page of Cheaper2Get | |
| Post-conditions | Suggested proposal(s) search result is displayed | |
| Use Case Scenarios | Normal Flow | * User click on Proposal search button * User enters search phrase * Systems searches the phrase and displays result. * User views group search result |
| Alternative Flows | (none) |
| Exceptions | * No result if no matches found |
| Includes | (none) |
| Rules | * Search result should include information on item description, number of voters, special offers(if any), … |
| Special Requirements | * Proposal Search takes less than 2 seconds. |
| Priority | 4 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC4 | |
| Use Case Name | Make Proposal | |
| User Story | As a user, I want to Make Proposal so that I can order an Item | |
| Actors | Customer | |
| Use Case Description | User makes a proposal in order to order an item | |
| Preconditions | Successful login | |
| Post-conditions | A new proposal (with detail description) is displayed | |
| Use Case Scenarios | Normal Flow | * User click on make Proposal button * User fills “New proposal” form * System verifies the input data and confirms the user * User sees the created proposal on the list |
| Alternative Flows | (none) |
| Exceptions | * Incorrect input * Conflict in proposal name; In case the new proposal name has the same name as another one in the list. |
| Includes | UC2 |
| Rules | (none) |
| Special Requirements | (none) |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC5 | |
| Use Case Name | Vote | |
| User Story | As a user, I want to Vote a proposal so that I can buy an item | |
| Actors | Customer | |
| Use Case Description | User votes a proposal | |
| Preconditions | Successful login | |
| Post-conditions | User voted a proposal | |
| Use Case Scenarios | Normal Flow | * User selects a proposal from the list * User sets quantity and clicks on “vote” button * User waits for confirmation message * User successfully votes an item. |
| Alternative Flows | (none) |
| Exceptions | (none) |
| Includes | UC4 |
| Rules | (none) |
| Special Requirements | (none) |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC6 | |
| Use Case Name | Pay | |
| User Story | As a user, I want to pay for the voted item so that I can get the item | |
| Actors | Customer | |
| Use Case Description | User pays for the voted items | |
| Preconditions | Successful login  User already voted for an item(s) | |
| Post-conditions | User successfully made a payment | |
| Use Case Scenarios | Normal Flow | * User selects a voted proposal from the list * User selects payment mode * User successfully makes a payment. |
| Alternative Flows | (none) |
| Exceptions | (none) |
| Includes | UC10 |
| Rules | * Payment has to be made with 24 hours after user votes for an item(s). |
| Special Requirements | * Payment has to be made with 24 hours after user votes for an item(s). |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC7 | |
| Use Case Name | Spread | |
| User Story | As a user, I want to Spread about proposal so that I can share information to friends and/or relatives | |
| Actors | Customer | |
| Use Case Description | User share proposal information to friends and/or relatives | |
| Preconditions | Successful login | |
| Post-conditions | User share proposal information to friends and/or relatives | |
| Use Case Scenarios | Normal Flow | * User selects a proposal from the list * User clicks on Spread button * User selects social account and clicks on “Send” button * System confirms the user for successful delivery. |
| Alternative Flows | (none) |
| Exceptions | * Wrong password social account |
| Includes | (none) |
| Rules | (none) |
| Special Requirements | (none) |
| Priority | 3 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC8 | |
| Use Case Name | View Status | |
| User Story | As a user, I want to check the status of my order so that I can know when to receive the item(s) | |
| Actors | Customer | |
| Use Case Description | User checks the status of his/her order | |
| Preconditions | Successful login  User already had ordered an item(s) | |
| Post-conditions | User Knows the status of his/her order | |
| Use Case Scenarios | Normal Flow | * User selects a proposal from the list * User clicks on “View Status” button * System shows the status of user’s order request. |
| Alternative Flows | (none) |
| Exceptions | (none) |
| Includes | UC5 |
| Rules | (none) |
| Special Requirements | (none) |
| Priority | 4 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC9 | |
| Use Case Name | Evaluate Proposal | |
| User Story | As a user, I want to Evaluate Users’ Proposal so that I can accept or reject it. | |
| Actors | SaaS Operator | |
| Use Case Description | User evaluates user proposals | |
| Preconditions | Successful login  Customer make a proposal | |
| Post-conditions | Proposal is evaluated and communicated to customers | |
| Use Case Scenarios | Normal Flow | * User selects a proposal from the list * User evaluates the proposal |
| Alternative Flows | (none) |
| Exceptions | (none) |
| Includes | UC4 |
| Rules | * Number of voters should be more than the minimum number |
| Special Requirements | * Evaluation should be done with three days once a proposal is created. |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |
| --- | --- |
| Supporting Requirement Model | Agile Software Requirements Model |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC10 | |
| Use Case Name | Approve Proposal | |
| User Story | As a user, I want to Approve Proposal so that I can deal with vendors. | |
| Actors | SaaS Operator | |
| Use Case Description | User approves user proposals | |
| Preconditions | Successful login  SaaS operator evaluates a proposal | |
| Post-conditions | Proposal is approved | |
| Use Case Scenarios | Normal Flow | * User selects a proposal from the list * User approves the proposal |
| Alternative Flows | (none) |
| Exceptions | (none) |
| Includes | UC9 |
| Rules | (none) |
| Special Requirements | (none) |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC11 | |
| Use Case Name | Deliver | |
| User Story | As a user, I want to deliver an item(s) so that I can finalize the deal | |
| Actors | SaaS Operator | |
| Use Case Description | SaaS Operator delivers item(s) | |
| Preconditions | Successful login  Payment has been made by a customer | |
| Post-conditions | Item is delivered | |
| Use Case Scenarios | Normal Flow | * SaaS operator selects an order * SaaS operator checks for payment * SaaS Operator collects the item(s) from vendor company * SaaS Operator delivers the item(s) |
| Alternative Flows | (none) |
| Exceptions | (none) |
| Includes | UC12 |
| Rules | (none) |
| Special Requirements | (None) |
| Priority | 5 | |
| Supporting Requirement Model | Agile Software Requirements Model | |

## Non-Functional Requirements

To make a good software, our group has identified top 5 quality attributes based on priority estimated from our product’s properties.

Table . Quality Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Quality Attribute | Requirement | Achievement Method | Priority |
| QA1 | Performance | Our software should operate in real-time(It’s response time should be less than 5 seconds) | Service Level Agreement  Performance Testing using AppNeta | 5 |
| QA2 | Availability | Our software should not have any down-time (It should be available 24/7) | Exhaustive Testing: Unit, Integration, System, Load, and Regression Test  Fall-over capabilities | 5 |
| QA3 | Usability | Our software should have an attractive and friendly user interface, and minimize the numbers of user-clicks for each function as much as possible | User survey & feedback | 5 |
| QA4 | Portability | Our software should be platform and OS independent: It should work on both mobile and PC, as well as in multiple operating systems including Windows and MacOS | Standard-based development: Web Service | 4 |
| QA5 | Security | Our software should prevent user information by limiting the access level of users as per their privilege | Security Testing (e.g. using CloudInspect tool) | 3 |
| QA6 | Reliability | Cheaper2Get should be reactive , hence, errors committed by users will be handled properly and appropriate confirmation message is sent to the user |  | 5 |
| QA7 | Supportability | Cheaper2Get should have ‘Help’ tab that gives general guides to users. |  |  |

## Customer Training and Support Requirements

As our company grows to be more matured, as a goal of Customer training and support, the below listed Application Maintenance Services(AMS) model used will be used:

**Application Support**

* Root Cause Analysis
* Enhancements
* New Developments
* Software Release Management
* Apply Patches

**User Support**

* Troubleshooting
* Answer “How to Question”
* Change Request

**Operation Support**

* Capacity Planning
* System Administration
* Database Administration
* Network Administration

**End User Support**

* User Manual
* Training Materials
* Knowledge Portal-online help
* Quick Reference Cards

Figure 5.. Customer Training and Support