Use Cases

# Sapientia Canteen MVC

*Roland Bogosi  
Computer Engineering III*

Table of Contents

[Sapientia Canteen MVC 1](#_Toc422445702)

[Login 3](#_Toc422445703)

[Registration 3](#_Toc422445704)

[Recovery 4](#_Toc422445705)

[Reservation 4](#_Toc422445706)

[Pass Payment 5](#_Toc422445707)

[Administration 5](#_Toc422445708)

# Login

|  |  |
| --- | --- |
| **Name** | Login |
| **Summary** | User specifies their username or email address alongside their password, and the web application authenticates them. |
| **Dependencies** | User has to be registered before signing in. |
| **Actors** | User, Administrator |
| **Preconditions** | Login form has to be displayed correctly and filled with the proper security token. |
| **Triggers** | User clicks on the login menu, or is redirected to the form by trying to access a page only intended for authenticated users. |
| **Parameters** | Username/Email, Password, CSRF Security Token |
| **Description** | **Standard Procedure**  The user opens the login form and fills it out with the correct information. If information is correct, the systems authenticates the user, otherwise an error is shown. After login, all the functions will be available to the user to which they are authorized to use.  **Special Exceptions**   * User specified incorrect username, email or password. * User is not registered. * Security token failed to validate. |
| **Postconditions** | User is logged in. |
| **Non-Functional Requirements** | Login form has to be served over HTTPS, as it handles sensitive information. |

# Registration

|  |  |
| --- | --- |
| **Name** | Registration |
| **Summary** | User specifies their username or email address alongside their password, and the web application registers them. |
| **Dependencies** | User has to be able to receive cookies and not be logged in currently. |
| **Actors** | User, Administrator |
| **Preconditions** | Register form has to be displayed correctly and filled with the proper security token. |
| **Triggers** | User clicks on the register menu, or is redirected to the form by trying to access a page only intended for authenticated users. |
| **Parameters** | Username/Email, Password, Captcha, CSRF Security Token |
| **Description** | **Standard Procedure**  The user opens the registration form and fills it out with the correct information. If information is correct, the systems registers the user, otherwise an error is shown. After registration, all the functions will be available to the user to which they are authorized to use, and the username is now reserved to the user for future use.  **Special Exceptions**   * User specified incorrect username, email or password. * Captcha is not filled out correctly. * Security token failed to validate. |
| **Postconditions** | User is registered and logged in. |
| **Non-Functional Requirements** | Registration form has to be served over HTTPS, as it handles sensitive information. |

# Recovery

|  |  |
| --- | --- |
| **Name** | Recovery |
| **Summary** | User specifies their username or email address, and the web application resets their password. |
| **Dependencies** | User has to be able to receive emails and not be logged in currently. |
| **Actors** | User, Administrator |
| **Preconditions** | Recovery form has to be displayed correctly and filled with the proper security token. |
| **Triggers** | User clicks on the recovery link from the Login or Register form. |
| **Parameters** | Username/Email, Captcha, CSRF Security Token |
| **Description** | **Standard Procedure**  The user opens the recovery form and fills it out with the correct information. If information is correct, the systems resets the user’s password via an email, otherwise an error is shown. After recovery, all the functions will be available to the user to which they are authorized to use.  **Special Exceptions**   * User specified incorrect username or email. * Captcha is not filled out correctly. * Security token failed to validate. |
| **Postconditions** | User is logged in. |
| **Non-Functional Requirements** | Recovery form has to be served over HTTPS, as it handles sensitive information. |

# Reservation

|  |  |
| --- | --- |
| **Name** | Reservation |
| **Summary** | User reserves their food for the full or partial week. |
| **Dependencies** | User has to be registered and logged in. |
| **Actors** | User, Administrator |
| **Preconditions** | Reservation form has to be filled out correctly such as that there are no more than one items selected per category per day. |
| **Triggers** | User clicks on the Week menu. |
| **Parameters** | Food Radio Boxes, CSRF Security Token |
| **Description** | **Standard Procedure**  The user opens the reservation form and checks all the food they want to reserve for themselves on the specific days. If information is correct, the systems saves this information, otherwise an error is shown. After reservation, the foods are reserved for the user at the canteen.  **Special Exceptions**   * Invalid or conflicting food ID selected/sent. * Security token failed to validate. |
| **Postconditions** | User’s foods are now reserved for the week. |
| **Non-Functional Requirements** | Reservation form has to be served over HTTPS, as it has semi-sensitive information. |

# Pass Payment

|  |  |
| --- | --- |
| **Name** | Pass Payment |
| **Summary** | User selects a pass and pays for it using a credit card. |
| **Dependencies** | User has to be registered, logged in, and own a credit card. |
| **Actors** | User, Administrator |
| **Preconditions** | Payment form has to be displayed correctly and filled with the proper security token. |
| **Triggers** | User clicks on the Pass menu and then on the Pay button near a pass. |
| **Parameters** | Credit Card, Expiration Date, CVV2, Stripe Security Token |
| **Description** | **Standard Procedure**  The user opens the payment form and fills it out accordingly. If information is correct, the systems saves this information and initiates a credit card charge, otherwise an error is shown. After payment, the pass is associated to the user’s account and they can begin using it at the canteen.  **Special Exceptions**   * Invalid credit card information. * Stripe token not valid. |
| **Postconditions** | User’s account has a specific pass associated to it. |
| **Non-Functional Requirements** | Payment form has to be served over HTTPS, as it handles sensitive information. |

# Administration

|  |  |
| --- | --- |
| **Name** | Administration |
| **Summary** | Administrator has full managerial control over the database. |
| **Dependencies** | User has to be registered, logged in, and have administrator privileges. |
| **Actors** | Administrator |
| **Preconditions** | CRUD forms have to be displayed correctly and filled with the proper security token. |
| **Triggers** | User clicks on the Administration menu, only available to users with administrator privileges. |
| **Parameters** | *All the Database Tables and their Columns* |
| **Description** | **Standard Procedure**  The user opens the administration page, selects a table to manage and initiates a change. If information is correct, the systems saves this information, otherwise an error is shown. After administration, the fresh data will be displayed on the webpage.  **Special Exceptions**   * Invalid table, column or ID in use. * Security token failed to validate. |
| **Postconditions** | Website’s data is updated. |
| **Non-Functional Requirements** | Administration forms have to be served over HTTPS, as they handle sensitive information. |