|  |  |
| --- | --- |
| **Use Case:** | Login |
| **Base Use Cases:** | N/A |
| **Primary Actors:** | Client – Restaurant admin – Delivery pilot |
| **Secondary actors** | N/A |
| **Trigger** | The user clicks Login button |
| **Preconditions:** | User must have registered |
| **Successful end conditions:** | Dashboard is opened showing available services. |
| **Failure end conditions:** | Account is not valid – error in backend server accessibility. |
| **Extensions:** | Required information missing. |
| **Included use cases:** | Check Validity |
| **Main  Success Scenario:**  **Extended Scenario:** | Step: Action   1. User clicks on login button. 2. The application shows up login form. 3. User fills in the login form and presses login. 4. Include :: Check Validity   User’s information details are checked.   1. Dashboard is opened showing available services.   Step: Branching Action   * 1. User information details are missing required details.   2. The application shows up warning message of the error.   3. User presses Try again.   4. Login form is showed up again. |

|  |  |
| --- | --- |
| **Use Case:** | Login as a Client |
| **Base Use Cases:** | Login |
| **Primary Actors:** | Client |
| **Secondary actors** | N/A |
| **Trigger** | The user clicks Login button |
| **Preconditions:** | User must have registered |
| **Successful end conditions:** | Dashboard is opened showing available services. |
| **Failure end conditions:** | Inserted information is not valid – error in backend server accessibility. |
| **Extensions:** | Required information missing. |
| **Included use cases:** | Check Validity |
| **Main  Success Scenario:**  **Extended Scenario:** | Step: Action   1. User clicks on login button. 2. The application shows up login form. 3. User fills in the login form and presses login. 4. Include :: Check Validity   **User’s information details are checked and checks account type.**   1. **Dashboard is opened showing available services for a client user**.   Step: Branching Action   * 1. User information details are missing required details.   2. The application shows up warning message of the error.   3. User presses Try again.   4. Login form is showed up again. |

|  |  |
| --- | --- |
| **Use Case:** | Login as a restaurant admin |
| **Base Use Cases:** | Login |
| **Primary Actors:** | Restaurant admin |
| **Secondary actors** | N/A |
| **Trigger** | The user clicks Login button |
| **Preconditions:** | User must have registered |
| **Successful end conditions:** | Dashboard is opened showing available services. |
| **Failure end conditions:** | Inserted information is not valid – error in backend server accessibility. |
| **Extensions:** | Required information missing. |
| **Included use cases:** | Check Validity |
| **Main  Success Scenario:**  **Extended Scenario:** | Step: Action   1. User clicks on login button. 2. The application shows up login form. 3. User fills in the login form and presses login. 4. Include :: Check Validity   **User’s information details are checked and checks account type.**   1. **Dashboard is opened showing available services for a restaurant admin user**.   Step: Branching Action   * 1. User information details are missing required details.   2. The application shows up warning message of the error.   3. User presses Try again.   4. Login form is showed up again. |

|  |  |
| --- | --- |
| **Use Case:** | Login as a delivery pilot |
| **Base Use Cases:** | Login |
| **Primary Actors:** | Delivery pilot |
| **Secondary actors** | N/A |
| **Trigger** | The user clicks Login button |
| **Preconditions:** | User must have registered |
| **Successful end conditions:** | Dashboard is opened showing available services. |
| **Failure end conditions:** | Inserted information is not valid – error in backend server accessibility. |
| **Extensions:** | Required information missing. |
| **Included use cases:** | Check Validity |
| **Main  Success Scenario:**  **Extended Scenario:** | Step: Action   1. User clicks on login button. 2. The application shows up login form. 3. User fills in the login form and presses login. 4. Include :: Check Validity   **User’s information details are checked and checks account type.**   1. **Dashboard is opened showing available services for a delivery pilot user**.   Step: Branching Action   * 1. User information details are missing required details.   2. The application shows up warning message of the error.   3. User presses Try again.   4. Login form is showed up again. |

|  |  |
| --- | --- |
| **Use Case:** | Check validity |
| **Base Use Cases:** | N/A |
| **Primary Actor:** | N/A |
| **Secondary actors** | N/A |
| **Trigger** | User added login information and pressed login |
| **Preconditions:** | Login form is all filled up |
| **Successful end conditions:** | User information is verified, and type of account is identified. |
| **Failure end conditions:** | User information is missing required information or not verified. |
| **Extensions:** | Required information missing |
| **Included use cases:** | N/A |
| **Main  Success Scenario:**  **Extended Scenario:** | Step: Action   1. User inserted information are provided to the system. 2. User credential database verifies the information inserted, determines the type of the account. 3. Information is returned as verified by the database and account type is returned to provide convenient services.   Step: Branching Action   * 1. User credential database doesn’t verify the inserted information.   2. The information is returned unverified. |

|  |  |
| --- | --- |
| **Use Case:** | Required information missing |
| **Base Use Cases:** | N/A |
| **Primary Actor:** | N/A |
| **Secondary actors** | N/A |
| **Trigger** | User inserted credentials is checked and not verified |
| **Preconditions:** | User credentials is checked |
| **Successful end conditions:** | User information is unverified. |
| **Failure end conditions:** | N/A. |
| **Extensions:** | N/A |
| **Included use cases:** | N/A |
| **Main  Success Scenario:** | Step: Action   1. User inserted information are defined as unverified. 2. User credential is checked if a required information is missing. |

|  |  |
| --- | --- |
| **Use Case:** | View to order |
| **Base Use Cases:** | N/A |
| **Primary Actor:** | Client |
| **Secondary actors** | N/A |
| **Trigger** | User logs in. |
| **Preconditions:** | User successes to login as a client. |
| **Successful end conditions:** | Application shows available convenient content that allows client to order. |
| **Failure end conditions:** | Application fails to show content. |
| **Extensions:** | N/A |
| **Included use cases:** | N/A |
| **Main  Success Scenario:** | Step: Action   1. User success to login as a client. 2. Application open a dashboard specially formed for client user. 3. Dashboard initially contains four of dishes of the day and four of restaurants of the day, options to select view of nearby restaurants, set or adjust a diet program. |

|  |  |
| --- | --- |
| **Use Case:** | View by dishes |
| **Base Use Cases:** | View to order |
| **Primary Actor:** | Client |
| **Secondary actors** | N/A |
| **Trigger** | User presses dishes option. |
| **Preconditions:** | User successes to login as a client. |
| **Successful end conditions:** | Application shows available dishes. |
| **Failure end conditions:** | Application fails to find dishes. |
| **Extensions:** | N/A |
| **Included use cases:** | N/A |
| **Main  Success Scenario:** | Step: Action   1. User presses dishes option. 2. Application searches for dishes available to be ordered. 3. Application finds a number of dishes. 4. Application shows dishes arranged by most frequently succeed to deliver orders, with a picture for each dish provided by restaurant admin and rate given by clients already received that specific dish and its price. Application provides an option to filter shown dishes by price, calories, type. |

|  |  |
| --- | --- |
| **Use Case:** | View by restaurants |
| **Base Use Cases:** | View to order |
| **Primary Actor:** | Client |
| **Secondary actors** | N/A |
| **Trigger** | Client user presses view by restaurants option. |
| **Preconditions:** | User succeed to login as a client. |
| **Successful end conditions:** | Application shows restaurants available. |
| **Failure end conditions:** | Application fails to find restaurants. |
| **Extensions:** | N/A |
| **Included use cases:** | N/A |
| **Main  Success Scenario:** | Step: Action   1. User presses view by restaurants option. 2. Application searches for restaurants available to serve. 3. Application finds a number of nearby restaurants. 4. Application shows restaurants arranged by most frequently succeed to deliver orders, with a picture or a logo for each restaurant provided by restaurant admin and rate given by clients already received orders. |

|  |  |
| --- | --- |
| **Use Case:** | Nearby restaurants |
| **Base Use Cases:** | View to order |
| **Primary Actor:** | Client |
| **Secondary actors** | N/A |
| **Trigger** | Client user presses nearby restaurants option. |
| **Preconditions:** | User succeed to login as a client. |
| **Successful end conditions:** | Application shows restaurants nearby user’s location. |
| **Failure end conditions:** | Application fails to find restaurants nearby user’s location. |
| **Extensions:** | N/A |
| **Included use cases:** | N/A |
| **Main  Success Scenario:** | Step: Action   1. User presses nearby restaurants option. 2. Application searches for restaurants near to user’s location. 3. Application finds a number of nearby restaurants. 4. Application shows restaurants arranged by most frequently succeed to deliver orders, with a picture or a logo for each restaurant provided by restaurant admin and rate given by clients already received orders. |