# Use Case Information

|  |  |
| --- | --- |
| Use Case # | UC-01 |
| Use Case Name | Check in with Foursquare |
| Description | **Example:** User checks in with Foursquare with their phone and the users’ location is changes in <System> accordingly. |

# Actor(s)

|  |  |
| --- | --- |
| Role/System Name | User Type and System Definition(s) |
| <System> | The prototype system that handles smart-spaces |
| Foursquare | Location-based social networking service for mobile devices |
| User | User that uses both Foursquare and <System> |

# Assumption(s)

|  |
| --- |
| <System> is functional |
| Foursquare is functional |
| <System> is a authenticated with Foursquares Real-time API |
| There is a constant Internet connection for all devices and services |

# Pre-Condition(s)

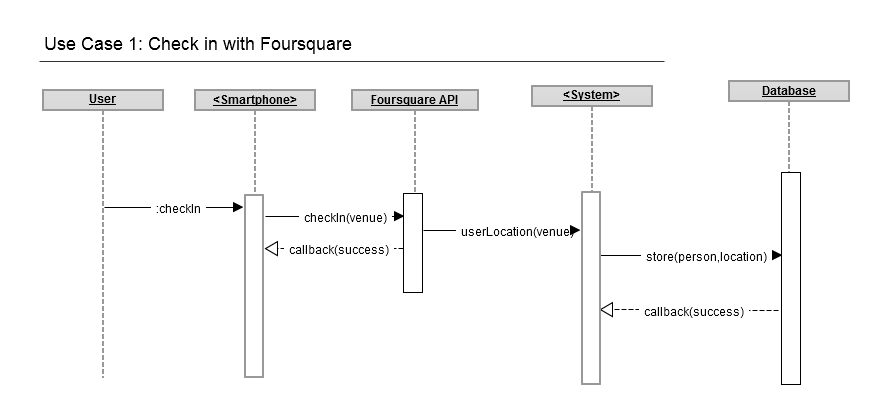
|  |
| --- |
| User is authenticated with Foursquare |
| User is a <System> user |
| User has linked Foursquare to <System> |

# Main case (Happy Path)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Use case | Data |
| 1 | User checks in a venue |  |  |  |
| 2 | Foursquare receives check-in info |  |  | Check-in data |
| 3 | Foursquare sends a POSTs to <System> |  |  |  |
| 4 | <System> gets the check-in info |  | UC-02 | Check-in data |
| 5 | <System> updates users current location | A-01 |  |  |

# Alternative Path(s)

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-01.1 | <System> does not update current position |  |  |



# Use Case Information

|  |  |
| --- | --- |
| Use Case # | UC-02 |
| Use Case Name | <System> gets new current location data |
| Description | **Examples:** User adds a calendar entry for the current time or a user checks-in via Foursquare |

# Actor(s)

|  |  |
| --- | --- |
| Role/System Name | User Type and System Definition(s) |
| <System> | The prototype system that handles smart-spaces |
| <LocationService> | Service or software which incorporates location information |
| User | User that uses both <LocationService> and <System> |

# Assumption(s)

|  |
| --- |
| <System> is functional |
| <LocationService> is functional |
| <System> and <LocationService> “knows” each other |
| There is a constant Internet connection for all devices and services |

# Pre-Condition(s)

|  |
| --- |
| User is authenticated with <LocationService> |
| User is a <System> user |
| User has linked <LocationService> to <System> |

# Main case (Happy Path)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Use case | Data |
| 1 | User uses <LocationService> to add a new current location |  |  |  |
| 2 | <LocationService> sends data to <System> | A-01 |  | Location data |
| 3 | <System> extracts location from ”Location data” | E-01 |  | Location data |
| 4 | <System> saves new location data in an RDF-store as location\_<LocationService> |  |  |  |
| 5 | <System> changes the users current location | A-02 |  |  |

# Alternative Path(s)

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-01.1 | <LocationService> stores “Location data” |  | Location data |
| A-01.2 | <System> periodically retrieves data from <LocationService> | E-01 | LocationService data |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-02.1 | <System> already has a location for the current time |  | current\_location |
| A-02.2 | <System> chooses the most current location update |  |  |
| A-02.3 | <System> changes the users current location |  |  |

# Exception Path(s)

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| E-01.1 | “Location data” is not present |  | Location data |
| E-01.2 | Keep current location data at previous location |  | current\_position |

# Use Case Information

|  |  |
| --- | --- |
| Use Case # | UC-03 |
| Use Case Name | Facial recognition with smartphone |
| Description | **Examples:** User uses the camera built-into or a 3rd-party camera connected to a smartphone and gets information about the people the camera is seeing |

# Actor(s)

|  |  |
| --- | --- |
| Role/System Name | User Type and System Definition(s) |
| <System> | The prototype system that handles smart-spaces |
| <Smartphone> | Phone with more advanced computing ability and connectivity than a feature phone. (Such as Android, iOS and Windows Phone devices) |
| User | User that uses both Foursquare and <System> |

# Assumption(s)

|  |
| --- |
| <System> is functional (as well as all data connections between internal <System> services) |
| <Smartphone> is functional |
| <Smartphone> is authenticated with <System> |
| There is a constant Internet connection for all devices and services |

# Pre-Condition(s)

|  |
| --- |
| User is a <System> user |
| User is authenticated with <System> |
| <Smartphone> had an application to communicate with <System> |
| User has linked <Smartphone>s application to <System> |

# Main case (Happy Path)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Use case | Data |
| 1 | User records a video/take image |  |  |  |
| 2 | <Smartphone> finds face(s) in the video/image | A-01 |  | Video data |
| 3 | <Smartphone> gets current GPS position | A-02 |  | GPS data |
| 4 | <Smartphone> gets the current date & time | A-03 |  | Datetime |
| 5 | <Smartphone> sends the image frame containing face(s), GPS data and date & time to <System> |  |  | Image frame, GPS data, Datetime |
| 6 | <System> receives data from <Smartphone> |  |  | SPhone data |
| 7 | <System> uses facial recognition algorithms on the image frame | E-01 |  | Video frame |
| 8 | Algorithms return User information | A-04 |  |  |
| 9 | <System> stores User information, GPS data and date & time in the database | A-05, E-02 |  | User info, GPS data, Datetime |
| 10 | <System> updates relations regarding the User, as well as the people in the image |  | UC-04 |  |

# Alternative Path(s)

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-01.1 | <Smartphone> does not find faces |  |  |
| A-01.2 | <Smartphone> waits for a new frame to process | E-03 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-02.1 | <Smartphone> does not get a GPS location |  |  |
| A-02.2 | <Smartphone> gets location from WLAN AP | A-06 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-03.1 | <Smartphone> does not have a set date & time |  |  |
| A-03.2 | <Smartphone> sets time to <System>\_time |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-04.1 | Facial recognition algorithm returns no faces |  |  |
| A-04.2 | <System> stores frame and face for user validation | A-07 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-05.1 | <System> stores the information available |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-06.1 | <Smartphone> sets location to “N/A” |  |  |

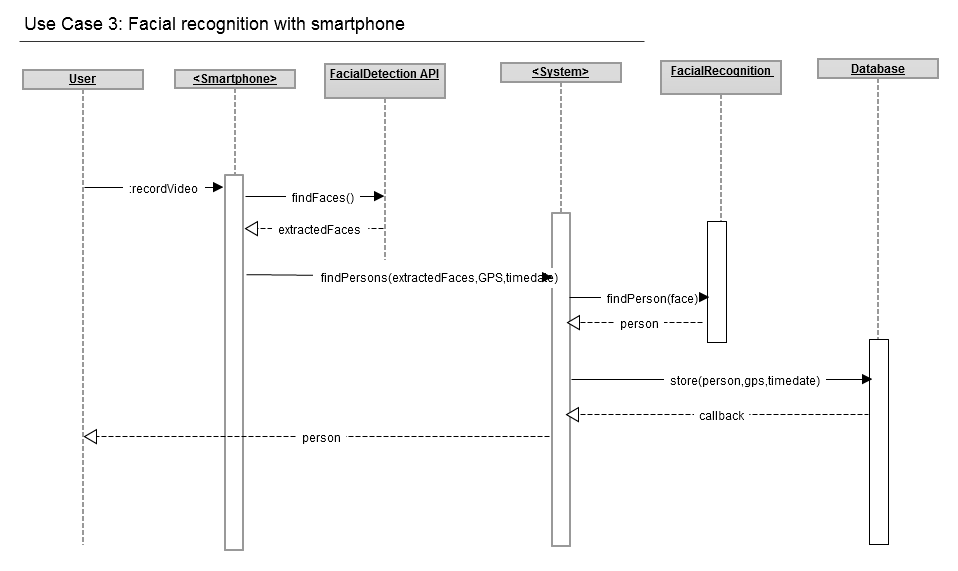
|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-07.1 | Image storage capacity full for specific user |  |  |
| A-07.2 | Replace the oldest image |  |  |

# Exception Path(s)

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| E-01.1 | Image type not supported error |  |  |
| E-01.2 | Stops the algorithms |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| E-02.1 | Date & time not found error |  |  |
| E-02.2 | Sets date & time to Null/None/Nil |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| E-03.1 | User closes application |  |  |



# Use Case Information

|  |  |
| --- | --- |
| Use Case # | UC-04 |
| Use Case Name | Update position from relation |
| Description | **Example:** The <System>s facial recognition algorithm finds a User (B) after getting an image frame from another User (A). (A) might also have specified a current location, which would put (B) at the same position as (A). |

# Actor(s)

|  |  |
| --- | --- |
| Role/System Name | User Type and System Definition(s) |
| <System> | The prototype system that handles smart-spaces |
| User A | User that uses <System>, and has supplied information to <System> |
| User B | User that uses <System>, and is mentioned in information supplied by User A |

# Assumption(s)

|  |
| --- |
| <System> is functional |

# Pre-Condition(s)

|  |
| --- |
| User A is a <System> user |
| User B is a <System> user |

# Main case (Happy Path)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Use case | Data |
| 1 | User A adds new information to the DB |  |  | new\_info |
| 2 | <System> finds location information regarding User B | A-01 |  | new\_info |
| 3 | <System> updates User B’s location from User A’s supplied information |  |  |  |

# Alternative Path(s)

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Step Description | Alternate & Exception Path | Data |
| A-01.1 | <System> does not find any information regarding User (B) |  |  |
| A-01.2 | <System> does not update current position for User (B) |  |  |