|  |
| --- |
| [Type the company name] |
| [Type the document title] |
| [Type the document subtitle] |

|  |
| --- |
| Y2P  [Pick the date] |

Contents

[1. INTRODUCTION 2](#_Toc495418408)

[A. Purpose: 2](#_Toc495418409)

[A.1 Goals 2](#_Toc495418410)

[B. Scope: 2](#_Toc495418411)

[C. Definitions, Acronyms, Abbreviations 3](#_Toc495418412)

[D. Revision history 3](#_Toc495418413)

[E. Reference Documents 3](#_Toc495418414)

[F. Document Structure 3](#_Toc495418415)

[2. OVERALL DESCRIPTION 3](#_Toc495418416)

[A. Product perspective: 3](#_Toc495418417)

[B. Product functions: 3](#_Toc495418418)

[C. User characteristics: 3](#_Toc495418419)

[D. Assumptions, dependencies and constraints: 3](#_Toc495418420)

[D.1. Domain Assumptions (BURAYA DAHA YAZILIR) 3](#_Toc495418421)

[D.2 Dependencies 4](#_Toc495418422)

[D.3 Constraints 4](#_Toc495418423)

[D.3.1 Regulatory Policies 4](#_Toc495418424)

[D.3.2 Hardware Limitations 4](#_Toc495418425)

[D.3.3 Interfaces to other applications 4](#_Toc495418426)

[3. SPECIFIC REQUIREMENTS: 4](#_Toc495418427)

[A. External Interface Requirements 5](#_Toc495418428)

[A.1 User Interfaces 5](#_Toc495418429)

[A.2 Hardware Interfaces 5](#_Toc495418430)

[A.3 Software Interfaces 5](#_Toc495418431)

[A.4 Communication Interfaces 5](#_Toc495418432)

[B. Functional Requirements: 5](#_Toc495418433)

[B.1 Use Cases 5](#_Toc495418434)

[C. Performance Requirements 11](#_Toc495418435)

[D. Design Constraints 11](#_Toc495418436)

[D.1 Standards compliance 11](#_Toc495418437)

[D.2 Hardware limitations 11](#_Toc495418438)

[D.3 Any other constraint 11](#_Toc495418439)

[E. Software System Attributes 11](#_Toc495418440)

[E.1 Reliability 11](#_Toc495418441)

[E.2 Availability 11](#_Toc495418442)

[E.3 Security 11](#_Toc495418443)

[E.4 Maintainability 11](#_Toc495418444)

[E.5 Portability 11](#_Toc495418445)

[4. FORMAL ANALYSIS USING ALLOY: 11](#_Toc495418446)

[5. EFFORT SPENT: 11](#_Toc495418447)

[6. REFERENCES 12](#_Toc495418448)

# 1. INTRODUCTION

A. Purpose:

This document presents the Requirement Analysis and Specifications for Travelander+ application. This document aims to define functional and non-functional requirements, detailed analysis of the environment and constraints, realization of customer needs, and characteristic of users and their use case for the Travelander+ application. Typical audience of this document are the developers who aims to implement this system.

Travelander+ application provides an enhanced calendar with custom scheduling options to create appointments and offers travel planning assistance in between the arranged appointments for its possible users. The system enables its users to add/edit appointments, choose mobility preferences and to offer them the best options for travelling from one appointment to another by regarding environmental conditions (such as traffic, weather etc.), user specific situations and preferences and efficient usage of time and other available resources.

### A.1 Goals

**Users:**

[G].Allow users to create new account

[G].Allow users to become logged in to existing account after entering his/her credentials.

[G].Allow users to view his/her calendar.

[G].Allow users to manage events in calendar

* [G].Allow users to add new event to the user if selected time slot is available.
* [G].Allow users to receive a warning if the selected time slot is occupied or not feasible by considering place and time constraints.
* [G].Allow users to edit his/her events.
* [G].Allow users to delete his/her events.
* [G].Allow users to add customized break time with a certain duration defined in a certain time interval
* [G].Allow users to pin particular event by regarding importance of the event.
* [G].Allow users to receive alerts for the pinned events.
* [G].Allow users to add periodic events in daily, weekly and monthly basis.

[G].Allow users to manage personal mobility preferences.

* [G].Allow users to enter personal mobility preferences.
* [G].Allow users to choose predefined mobility preferences such as preference lists enforcing minimizing carbon prints, not driving car, not using public transportation etc.
* [G].Allow users to activate or deactivate particular mobility options

[G].Allow users to know mobility options which minimize travelling duration under user preferences, weather and traffic constraints.

[G].Allow users to generate the optimum schedule in consideration of time and travelling resources.

B. Scope:

Travelander+ will be the mobile and web application that enables to manage appointments and find best mobility options for its users. Its users can be everyone who needs to plan his/her long or short term schedule. Since the system is able to take information from various sources such as maps, traffic analysis on Internet, weather forecasting etc., Travelander+ is able to adapt the appointments and mobility options for maximizing efficiency on time and minimizing the latency and usage of other resources.

## C. Definitions, Acronyms, Abbreviations

### C.1 Definitions

* Event: Any appointment or customized break by the user

### C.2 Acronyms

* RASD : Requirement Analysis and Specification Document
* PTIP: Public Transportation Information Provider
* API: Application Programming Interface

## D. Revision history

## E. Reference Documents

## F. Document Structure

This document consists of 6 parts as table of content indicates:

1. Introduction: The problem definition is introduced and aims of both this document and desired system are explained. Goals of the system are listed with mapping abbreviations. Basically, this part provides the introductory information for giving full understanding of rest of the document
2. Overall Description:
3. Specific Requirements: As the title indicates, this part lists all the functional and non-functional requirements of the system

# 2. OVERALL DESCRIPTION

## A. Product perspective:

here we include further details on the shared phenomena and a

domain model (class diagrams and statecharts)

## B. Product functions:

In this part, the main functions of the proposed system are introduced and explained. By providing these essential functions, the system aims to satisfy the main needs of the described problem within given boundaries of both environment and system itself.

### B.1 Event Management

Event management is one of the main functions of this system. It basically enables to user add/edit/delete appointments and customized breaks and, visualize and plan his/her schedule. By accomplishing these purposes, the event manager promises that;

* The user is never late for any appointment
* No overlapping event might occur
* Optimum efficiency of time management is always achieved.
* Travelling durations are also taken account for planning schedule.

With these given necessities of the system, this functionality of the system solves the event management part of the problem.

### B.2 Trip Management

This functionality gathers the most recent information and does the necessary calculations for offering best mobility options to its user by considering current location, weather and traffic situation, recent events and also user preferences and conditions. This functionality successfully able to communicate with external APIs for gathering information related to current location, maps, public transportation, weather forecasting, traffic etc. and merge them with user preferences and added personal events.

## C. User characteristics:

Only active actor for this system is the user; who aims to obtain optimized schedule planning, mobility options and calendar services. System requires from the user registration for his/her personal account at the first place. After registration, user is needed to input his/her appointments and personal preferences for the operation of main functions of the system.

## D. Assumptions, dependencies and constraints:

### D.1. Domain Assumptions

Our proposed system assumes that these assumptions hold for the domain that the system operates on:

1. During program working, operating device always receives Internet connection.
2. Information about weather forecasting and traffic conditions are published on Internet
3. Resources for the updated weather forecasting and traffic conditions always provide accurate information.
4. Information of public transportation is published on Public Transportation Information Provider(PTIP) API.
5. Public transportation vehicles are assumed as punctual with their published programs.
6. Provided information of ticket prices, ticket seller locations and working hours, and stop locations of public transportation are always accurate.
7. Users who prefer driving have already driver license for the preferred vehicle if it is necessary (for motorbike, car etc.)
8. Users are able to ride bike when biking is activated as mobility option.
9. Users are assumed to walk in average speed.
10. Users do not have any disability related to walking if walking is activated as mobility option.
11. During mobile application operation, GPS is on and at working status while current position is needed.
12. GPS API always provides accurate location position.
13. Each user account has only one calendar.
14. Users always know and enter correctly their credentials.
15. Usernames are unique and consists of only [A..Z],[a..z],[0..9] characters.
16. No user can be different places at the same time.
17. Users accurately enter location addresses and date-time of the events to the system.
18. Break duration is always equal or greater than given time interval for it.

### D.2 Dependencies

### D.3 Constraints

### D.3.1 Regulatory Policies

Proposed system requires user input as appointment time and location which are details of personal schedule and only uses this information for event and trip management purposes. Also, it asks for the user’s permission in case of the usage of current location is needed and it is only used for increasing the quality of trip planning. It is not served for commercial or any other purpose.

### D.3.2 Hardware Limitations

For the mobile application, user needs a device with:

* At least 3G Internet connection
* GPS Connection
* Compatible operating system and device (IOS or Android smartphone)
* Space for application

### D.3.3 Interfaces to other applications

System needs to communicate for up-to-date information collection of weather, traffic, public transportation. These operations are conducted with related APIs. Also, it needs to communicate and manage a database system for storing and updating personal user information such as credentials, recent schedule and events, preferences for mobility options. For this purpose, MySQL will be deployed

3. SPECIFIC REQUIREMENTS:Here we include more details on all aspects in Section 2 if they can

be useful for the development team.

## A. External Interface Requirements

### A.1 User Interfaces

### A.2 Hardware Interfaces

### A.3 Software Interfaces

### A.4 Communication Interfaces

## B. Functional Requirements:

### B.1.1 [G ]. Allow users to create new account

* [R ]. The system must initiate new user registration with user input. During this process, the system must ask username and password to the user.
* [R ]. The system must check whether the username are taken by another user or not
* [R ]. The system must check whether the username have written only with [A..Z] [a..z] [0..9] characters or not
* [D ]. Usernames are unique and consists of only [A..Z],[a..z],[0..9] characters.

### B.1.2 [G ]. Allow users to become logged in to existing account after entering his/her credentials.

* [R ]. The system must check whether the credentials are correct by querying in the database.
* [D ]. Users always know and enters correctly their credentials.

### B.1.3 [G ]. Allow users to view his/her calendar.

* [R ]. Each registered user must be able to log in to his/her personal account
* [R ]. The system must retrieve details the personal calendar of user from the database when it is required by the user.
* [D ]. Each user account has only one calendar.

### B.1.4 [G ]. Allow users to manage events in calendar

* [R ]. Each registered user must be able to log in to his/her personal account
* **[**R ]. The system must be able to add event to calendar by registered user input and must check validity of calendar by controlling whether the selected time slot is available or not due to any reason.
* [R ]. The system must be able to delete event from calendar by registered user input.
* [R ]. The system must be able to edit an existing event from calendar by registered user input and must check validity of edited calendar by controlling whether the selected time slot is available or not due to any reason.
* [R ]. The system must pin particular events by registered user input and must generate reminders or alarms for them.
* [R ]. The system must be able to add events as periodically in daily, weekly or monthly basis.
* [R ].

### B.2 Use Cases

|  |  |
| --- | --- |
| **Use Case 1** | Log in |
| **Actors:** | User |
| **Entry Condition:** | The user is signed up to the system. |
| **Flow of Events:** | 1. The user opens the Travlendar+ application on his/her device. 2. The system displays the main screen. 3. The user enters his/her e-mail address or user name and the predefined password. 4. The user selects log in. 5. The system displays the Mobility Suggestions for the day’s schedule. 6. The user selects OK. 7. The system displays tools menu with the profile information. |
| **Exit Condition:** | The use case terminates when the user is successfully logged in. |
| **Exceptions:** | * The user information entered by the user i.e. the email address/user name or the password is wrong hence an error message is displayed. |

|  |  |
| --- | --- |
| **Use Case 2** | Add Event |
| **Actors:** | User |
| **Entry Condition:** | The user is logged in. |
| **Flow of Events:** | 1. The user selects the My Events function from the Main Menu. 2. The user slides the screen down. 3. The system creates a new line on the top of the events list. 4. The user enters a name for the event. 5. The system directs the user to the calendar automatically. 6. The user selects the date that he/she wishes to add the event from the calendar view. 7. The user selects the event’s start and end time on the selected date. 8. The system directs the user to the map. 9. The user selects the event location by entering the specific location name or the address. 10. The user saves the event details. 11. Travlendar+ displays a confirmation message notifying the user that the event is added to the calendar. |
| **Exit Condition:** | * The use case terminates when the selected date and time is assigned to the new event. * The selected date and time interval is now unavailable to be allocated by any other new event to be added in the future |
| **Exception 1:** | * If the time interval on the date selected by the user is already assigned to another event, the system displays a warning message indicating that the desired date/time is unavailable. * The system displays another message after the warning asking the user if he/she would like to change the date/ time * If the user selects “Yes” the flow starts again from the second step. * If the user selects “No”, the use case terminates without any date/time allocation. |
| **Exception 2:** | * If the time interval on the date selected by the user is free but the location is unreachable for the selected time according to the system calculations based on the current schedule, the system displays a warning message indicating the event is unreachable. * The system displays another message after the warning asking the user if he/she would like to change the event location. * If the user selects “Yes” the flow starts again from the third step. * If the user selects “No”, the use case terminates without any date/time allocation. |

|  |  |
| --- | --- |
| **Use Case 3** | Create Repeated Event |
| **Actors:** | User |
| **Entry Condition:** | The user is logged in. |
| **Normal Flow of Events:** | 1. The user selects the My Events function from the Main Menu. 2. The user slides the screen down. 3. The system creates a new line on the top of the events list. 4. The user enters a name for the event. 5. The system directs the user to the calendar automatically. 6. The user selects the date that he/she wishes to add the event from the calendar view. 7. The user selects the event’s start and end time on the selected date. 8. The system directs the user to the map. 9. The user selects the event location by entering the specific location name or the address. 10. The user slides the map view to the left to see event details. 11. The system displays a list screen including the event details which are Name, Date/Time, Location, Duration and Frequency. 12. The user edits the frequency which is set to null as default based on the event’s nature (e.g. daily, weekly, monthly or customized (e.g. Weekly: Monday-Wednesday)). 13. The user may edit the duration which is set to the whole-time interval between the start and end time of the event as default based on the event’s nature (e.g. 45-minute lunch break between the selected start and end time). 14. The user saves the event details. 15. Travlendar+ displays a confirmation message notifying the user that the event is added to the calendar with the defined frequency. |
| **Exit Condition:** | * The use case terminates when, starting with the selected date, the selected time is duplicated according to the frequency, assigned to the event and added to the other dates on the calendar. * The selected dates and time intervals appearing on the calendar based on the frequency are now unavailable to be allocated by any other new event to be added in the future |
| **Alternative Flow of Events:** | 1. The user selects the My Events function from the Main Menu. 2. The user selects the existing event he/she would like to repeat. 3. The flow continues as in the normal flow starting from the fifth step. |
| **Exceptions:** | * The system displays a warning message indicating the event cannot be repeated since one or more date/time that needs to be allocated based on the frequency are unavailable. * The system displays another message after the warning asking the user if he/she would like to change the date/ time or frequency. * If the user selects “Yes” the flow starts again from the sixth step. * If the user selects “No”, the use case terminates without any date/time allocation in the normal flow or only allocating the existing single event date/time in the alternative flow. |

|  |  |
| --- | --- |
| **Use Case 4** | Edit Event Date/Time |
| **Actors:** | User |
| **Entry Condition:** | * The user is logged in. * The event list is not empty. |
| **Flow of Events:** | 1. The user selects the My Events function from the Main Menu. 2. The user selects the event he/she would like to edit. 3. The system directs user to the date selected on the calendar. 4. The user edits the date and/or time. 5. The system displays a message asking the user “Would you like to save the changes?”. 6. The user selects Yes. 7. Travlendar+ displays a confirmation message notifying the user that the changes are saved. |
| **Exit Condition:** | * The use case terminates when the new date and time is assigned to the event. * The new date and/or time interval is now unavailable to be allocated by any other new event to be added in the future. * The previous date and/or time assigned to the event is now available to be allocated by any other new event to be added in the future. |
| **Exceptions 1:** | * The user selects No after the system displays the “Would you like to save the changes?” message. * The use case terminates without changing the date and/or time of the event. * The event’s current date and time interval stays unavailable to be allocated by any other new event to be added in the future. |
| **Exceptions 2:** | * If the time interval on the new date and/or time selected by the user is already assigned to another event, the system displays a warning message indicating that the desired date/time is unavailable. * The system displays another message after the warning asking the user if he/she would like to change the date/ time. * If the user selects “Yes” the flow starts again from the fourth step. * If the user selects “No”, the use case terminates keeping the current date/time allocated. |

|  |  |
| --- | --- |
| **Use Case 5** | Edit Event Location |
| **Actors:** | User |
| **Entry Condition:** | * The user is logged in. * The event list is not empty. |
| **Flow of Events:** | 1. The user selects the My Events function from the Main Menu. 2. The user selects the event he/she would like to edit. 3. The system directs user to the date selected on the calendar. 4. The user slides the screen to the left. 5. The system displays the location on the map view. 6. The user selects a new location by entering the address or the specific location name. 7. The system displays a message asking the user “Would you like to save the changes?”. 8. The user selects Yes. 9. Travlendar+ displays a confirmation message notifying the user that the changes are saved. |
| **Exit Condition:** | * The use case terminates when the new location is assigned to the event. * The event’s current date and time interval stays unavailable to be allocated by any other new event to be added in the future. |
| **Exceptions 1:** | * The user selects No after the system displays the “Would you like to save the changes?” message. * The use case terminates without changing the location of the event. |
| **Exceptions 2:** | * If the new location is unreachable for the selected time according to the system calculations based on the current schedule, the system displays a warning message indicating the event is unreachable. * The system displays another message after the warning asking the user if he/she would like to select another event location. * If the user selects “Yes” the flow starts again from the sixth step. * If the user selects “No”, the use case terminates keeping the current location unaltered. |

|  |  |
| --- | --- |
| **Use Case 6** | Delete Event |
| **Actors:** | User |
| **Entry Condition:** | * The user is logged in. * The event list is not empty. |
| **Flow of Events:** | 1. The user selects the My Events function from the Main Menu. 2. The user holds the event he/she wants to delete and slides it to the left. 3. The system displays a message asking the user “Would you like to delete this event?”. 4. The user selects Yes. 5. The system removes the event from the list and the calendar. |
| **Exit Condition:** | * The use case terminates when the event is removed from the list and the calendar. * The deleted event’s date and time interval is now available to be allocated by any other new event to be added in the future. |
| **Exceptions:** | * The user selects No after the system displays the “Would you like to delete this event?” message. * The use case terminates without removing the event from the list or the calendar. * The event’s date and time interval stays unavailable to be allocated by any other new event to be added in the future. |

|  |  |
| --- | --- |
| **Use Case 7** | Sort Mobility Preferences |
| **Actors:** | User |
| **Entry Condition:** | * The user is logged in. * The mobility option is active. |
| **Flow of Events:** | 1. The user selects Mobility Option Preferences from the Main Menu. 2. The system displays all mobility options. 3. The user sorts the mobility options based on his/her preferences (e.g. if the user prefers to walk or ride a bike as much as possible he/she should put these options on the top of the list) by holding the option and swiping it to the desired row on the list or he/she can turn on the “Minimize Carbon Foot Print” option displayed at the end of the list and the options are automatically sorted to fulfill this goal. |
| **Exit Condition:** | * The use case terminates when the system saves the preferences to be used in the best mobility option algorithm. |

|  |  |
| --- | --- |
| **Use Case 8** | Deactivate Mobility Option |
| **Actors:** | User |
| **Entry Condition:** | The user is logged in. |
| **Flow of Events:** | 1. The user selects Mobility Option Preferences from the Main Menu. 2. The system displays all mobility options. 3. The user holds the mobility option he/she wants to deactivate and slides it to the left. 4. The system fades the deactivated option and puts it to the end of the list. |
| **Exit Condition:** | * The use case terminates when the system saves the mobility option’s state to be used in the best mobility option algorithm. * The customization and sorting options are unavailable for the deactivated mobility option. |
| **Exceptions:** |  |

|  |  |
| --- | --- |
| **Use Case 9** | Activate Mobility Option |
| **Actors:** | User |
| **Entry Condition:** | The user is logged in. |
| **Flow of Events:** | 1. The user selects Mobility Option Preferences from the Main Menu. 2. The system displays all mobility options. 3. The user holds the deactivated mobility option he/she wants to activate and slides it to the right. 4. The system brightens the activated option and leaves it at the end of the list. |
| **Exit Condition:** | * The use case terminates when the system saves the mobility option’s state to be used in the best mobility option algorithm. * The customization and sorting options are now available for the activated mobility option. |
| **Exceptions:** |  |

|  |  |
| --- | --- |
| **Use Case 10** | Customize Mobility Option/ Restrict Undesired Time Interval |
| **Actors:** | User |
| **Entry Condition:** | * The user is logged in. * The mobility option is active. |
| **Flow of Events:** | 1. The user selects Mobility Option Preferences from the Main Menu. 2. The system displays all mobility options. 3. The user selects the mobility option he/she wants to customize. 4. The user selects a start and end time from the “Restricted Time Interval” drop down menu. |
| **Exit Condition:** | * The use case terminates when the system saves the custom preference to be used in the best mobility option algorithm. |
| **Exceptions:** |  |

|  |  |
| --- | --- |
| **Use Case 11** | Customize Mobility Option/ Set Distance Limit |
| **Actors:** | User |
| **Entry Condition:** | * The user is logged in. * The mobility option is active. |
| **Flow of Events:** | 1. The user selects Mobility Option Preferences from the Main Menu. 2. The system displays all mobility options. 3. The user selects the mobility option he/she wants to customize. 4. The user enters the maximum distance limit he/she would utilize the selected mobility option. 5. The user selects the unit of measure from the drop down list. |
| **Exit Condition:** | * The use case terminates when the system saves the custom preference to be used in the best mobility option algorithm. |
| **Exceptions:** |  |

|  |  |
| --- | --- |
| **Use Case 12** | Edit Mobility Preferences |
| **Actors:** | User |
| **Entry Condition:** | The user is logged in. |
| **Flow of Events:** | 1. The user selects Mobility Option Preferences from the Main Menu. 2. The system displays the previously sorted and customized mobility options list. 3. The user edits his/her preferences (e.g. resorts the list, changes the customizations or activates a deactivated mobility option.) |
| **Exit Condition:** | * The use case terminates when the system saves the new preferences to be used in the best mobility option algorithm. |
| **Exceptions:** |  |

## C. Performance Requirements

## D. Design Constraints

### D.1 Standards compliance

### D.2 Hardware limitations

### D.3 Any other constraint

## E. Software System Attributes

### E.1 Reliability

### E.2 Availability

### E.3 Security

### E.4 Maintainability

### E.5 Portability

4. FORMAL ANALYSIS USING ALLOY:in this section you will include your Alloy model. We require

you to comment on the model by discussing the purpose of the model, what you can prove

with it and why what you prove is important given the problem at hand. You are also

required to show one or more worlds obtained by running your model.

5. EFFORT SPENT:In this section you will include information about the number of hours each

group member has worked for this document.

# 6. REFERENCES