

Requirement Specification

ECO-TRAVEL



Student Number	Name
2014314081	김주한
2015310965	류민재
2016314421	나빌
2015314230	선명우
2015310434	권태완

TABLE OF CONTENTS

1. Preface.....	5
1.1 READERSHIP	5
A. User Requirements Readership	5
B. User Requirements Readership	5
1.2. DOCUMENT STRUCTURE.....	6
A. Introduction.....	6
B. Glossary.....	6
C. User Requirements Definition.....	6
D. System Architecture	6
E. System Requirements Specification.....	6
F. System Model.....	7
G. System Evolution.....	7
H. Appendices	7
I. Index	7
2. Introduction	8
2.1. NEEDS.....	8
2.2 SYSTEM OVERVIEW	10
2.3 EXPECTED EFFECTS.....	11
A. 기존의 여행일정 실시간 공유 => 여행 참여도 UP	11
B. 카메라/사진 영수증 자동인식 => 예산, 지출내역 관리 편의성 UP	12
3. Glossary	13
4. User Requirements Definition	15
4.1 OBJECTIVE	15
4.2 FUNCTIONAL REQUIREMENTS	15
A. Sign Up & Login.....	15
B. Travel Schedule Management.....	16
C. Budget Management	17

Requirements Specification

D.	Receipt Camera Recognition	18
E.	Schedule Map	18
F.	Friends Invitation	19
4.3	NON-FUNCTIONAL REQUIREMENTS	20
A.	Product Requirements	20
B.	Organizational Requirements	21
C.	External Requirements	21
5.	System Architecture	23
5.1	FRONTEND ARCHITECTURE	23
5.2	SOCKET COMMUNICATION SYSTEM	24
5.3	BACKEND ARCHITECTURE	25
5.4	IMAGE RECOGNITION SYSTEM	26
5.5	REAL-TIME EXCHANGE RATE SYSTEM	27
6.	System Requirements Specification	28
6.1	OBJECTIVE	28
6.2	FUNCTIONAL REQUIREMENTS	28
A.	로그인	28
B.	회원가입	29
C.	새 여행 추가하기	29
D.	여행 목적지(국가, 도시) 검색	30
E.	여행 같이 갈 친구 검색하기	30
F.	여행목록 전체리스트 보여주기	31
G.	여행 상세보기	32
H.	여행일정 보여주기	32
I.	여행일정 추가(수정)하기	33
J.	예산관리 보여주기	33
K.	Show pay-map	34
L.	영수증 자동인식	34
6.3	NON-FUNCTIONAL REQUIREMENTS	36
A.	Product Requirements	36
B.	Organizational Requirements	37
C.	External Requirements	37

Requirements Specification

6.4 SCENARIO EXAMPLES	38
A. Add New Travel Schedule Scenario.....	38
B. Camera Receipt Recognition Scenario	39
7. System Models	41
7.1 CONTEXT MODELS.....	41
A. Context Diagram.....	41
B. Process Diagram.....	42
7.2 INTERACTION MODELS.....	43
A. Use Case Diagram	43
B. Tabular Description of Use Case Diagram	44
7.3 BEHAVIORAL MODELS	47
A. 영수증 자동인식 DFD.....	47
B. 실시간 환율적용 DFD.....	48
8. System Evolution	49
8.1 OBJECTIVE	49
8.2 EVOLUTIONS	49
A. Travel Recommendation	49
B. Search Nearby Companions	50
C. Automatically Setting Budget	50
D. Oauth Implementation.....	51
E. Kakao-talk Friends Invitation	52
F. Add Comment/Re-comment on schedule	52
9. Appendices.....	54
9.1 HARDWARE REQUIREMENTS	54
9.2 DATABASE REQUIREMENTS.....	54
A. User.....	55
B. Travel	55
C. Schedule.....	55
D. Budget	55
E. Travel-schedule mapping table.....	55
F. Nations/cities	55

Requirements Specification

10. Index	56
10.1 TABLES	56
10.2 FIGURES	57
10.3 DIAGRAMS	58
11. References.....	59

1. Preface

This chapter defines the expected readership of the document, and briefly introduces the content of each chapter. This chapter also describes version history including a rationale for creation of a new version and a summary of the changes made in each version.

1.1 Readership

This document is intended to be read by a variety of readers. Therefore, descriptions for each readership are explained.

A. User Requirements Readership

User requirement is written for customers. The chapter refrains from using specialized terminology to explain requirements from the user's point of view, and is described in natural language with visuals such as diagrams.

B. System Requirements Readership

A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

1.2. Document Structure

A. Introduction

Describe the needs of the system to be developed, outline of the system structure and function of the system. Also describes the goal and target in developing the system.

B. Glossary

Define technical terms used throughout the document and describe in what context the terms are used. This document includes not only system developers but also users and stakeholders as well, so the terminology is explained in detail.

C. User Requirements Definition

Describe the functional and non-functional requirements of the system from the user's point of view. In addition to brief schematics, the documentation is written using natural language to make the documentation easier for the system user.

D. System Architecture

Outlines of the structure of the system. Explains how the functions of the system are allocated and distributed to each subsystem.

E. System Requirements Specification

Based on the requirements outlined in the User Requirements Definition, the functional requirements, non-functional requirements, and other requirements are described in detail. A structured document setting out detailed descriptions of the

Requirements Specification

system's functions, services and operational constraints.

F. System Model

Relationship between each component of the system and the relationship between the external environment are explained with diagrams.

G. System Evolution

Describes the developmental limitations of the system, anticipates the change that may occur during the operation of the system, and describes the countermeasures.

H. Appendices

Describe any references and resources omitted from the main text. Includes things such as hardware, database requirements, and development environment.

I. Index

Describe the index of figures, tables, diagrams, etc. used in this document.

2. Introduction

This chapter describes the needs of the system to be developed, outline of the system structure and function of the system. Also describes the goal and target in developing the system.

2.1. Needs

Each year, the number of people who travel in the country increase year by year. According to the survey made by Korea Culture & Tourism Institute, the total number of domestic trips for year 2017 reach almost 300,000. This means that the demand for people going for vacations has never been higher.

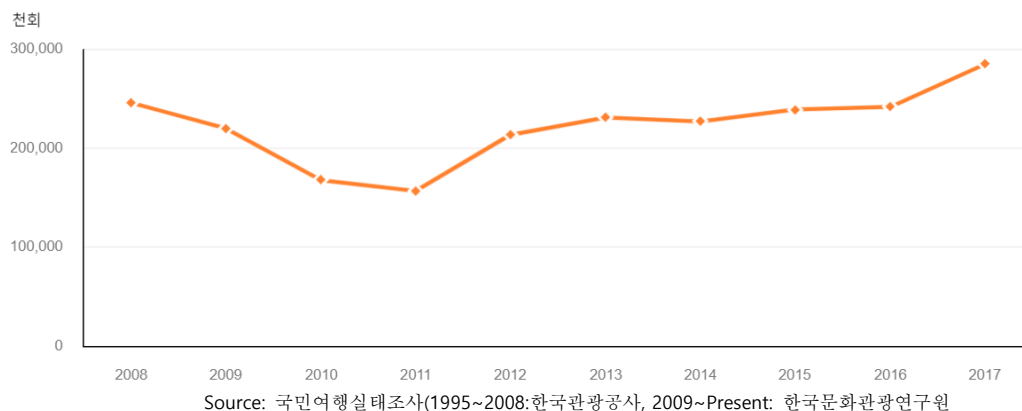
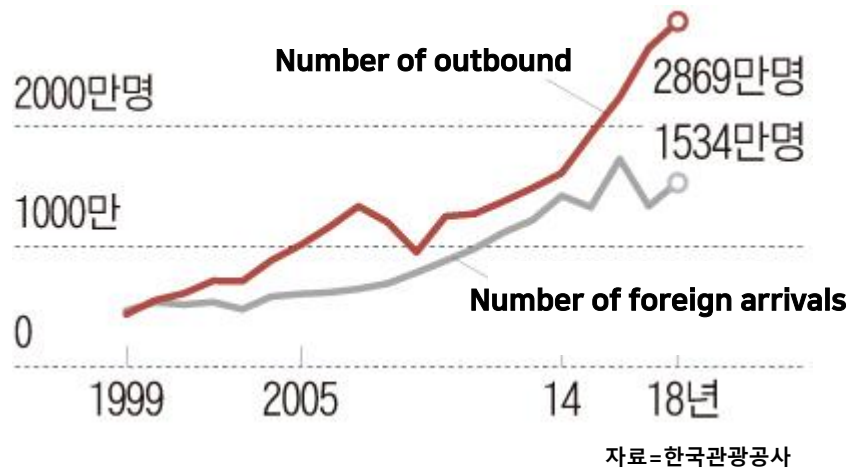


Figure 1: 국민여행 실태조사

According to Korean Statistical Information Service, the number of Koreans people going out of the country – by means of vacation also increase drastically in the year 2018 compared to 2017. This means that not only in terms of domestic travel, but in terms of international travel, people have been more interested and capable of going to vacations.

Number of outbound and foreign arrivals



This increase in the number in both statistics show that the demand for travelling is growing. More and more people are sharing their travel experience on social network sites such as Instagram, Facebook and Twitter.

To make things easier for people who travel, we need a travel application which can record and share our travel activities with friends in real time and also manage travel budget/expenditures. **Despite advances in photo-recognition technology, applications on the market still require users to manually fill in their budgets or expenditures.** Therefore, we are thinking of creating a travel diary in which you can manage and record budgets or expenditures of your trip in real time in an easier way than applications on the market. You can manage and records the history of your travel. Also, You can make consumption records more intuitive by adapting OCR technology so that consumption can be controlled better and more rationally when you turn back to the travel or another travel. We can write a travel diary whenever we finished that wonderful experience, whenever we can quickly find what we wrote and enjoy that good memories.

2.2 System overview

Eco-travel 은 OCR기술과 딥러닝 기술을 기반으로 영수증 사진을 찍기만 하면, 자동으로 지출 내역이 기입되어 여행 예산, 지출을 관리할 수 있는 시스템이다. 이와 더불어, 여행 스케줄관리 기능, 이를 지도에 표시할 수 있는 일정지도를 표시하여, 여행자들의 일정관리의 편의성을 증진시키고자 한다. 또한, 총무, 기록을 맡는 사람뿐만 아닌, 모두가 일정과 지출내역을 공유할 수 있는 시스템을 제공한다.

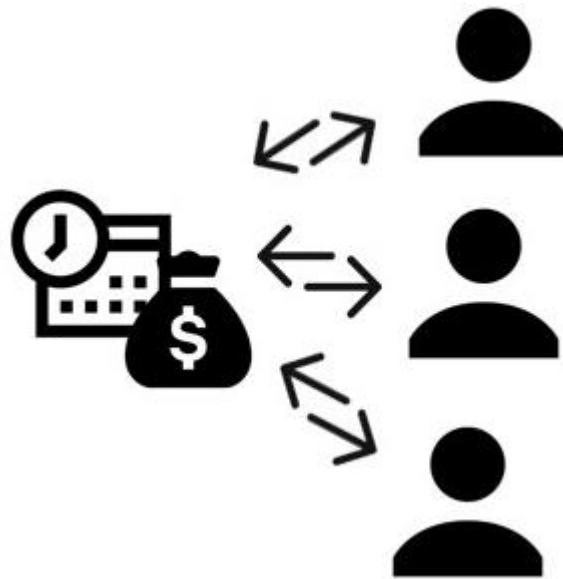


Figure 3: 시스템 목업 - 일정 실시간 공유

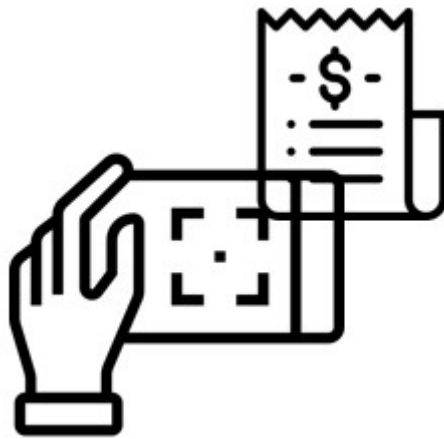


Figure 4: 시스템 목업 - 영수증 카메라 자동인식

Eco-travel 은 언제나 사용자 지향적이며, 여행자들의 불편을 개선하는 데 힘쓸 것입니다. 이와같은 기능들을 갖춘 Eco-travel 시스템은 여행자들의 완벽한 일정관리 어플리케이션이 되는 것을 지향점으로 삼고 있습니다.

2.3 Expected Effects

A. 기존의 여행일정 실시간 공유 => 여행 참여도 UP

기존의 여행일정관리 어플리케이션은 대다수가 기록을 하는 당사자 본인만이 일정을 관리할 수 있었다. 불특정 다수의 타인에게 자신의 일정을 알리는 데는 인스타그램을 활용하는 것이 가장 효과적이지만, 일정과 더불어 예산관리도 같이 해야하는 니즈가 있는 여행객들에게 인스타그램은 온전한 여행관리 시스템이 될 수 없다. 따라서, 같이 여행을 하는 동행객들끼리 실시간으로 일정과 예산, 지출내역을 공유할 수 있는 시스템은 기존의 한 명의 총무가 일정과 예산을 담당하여, 비교적 나머지 동행객들의 여행 참여도가 낮은 문제를 해결해 줄 것이다.

Requirements Specification

B. 카메라/사진 영수증 자동인식 => 예산, 지출내역 관리 편의성 UP

시중에 나와있는 여행예산관리 시스템은 9할 이상이 유저가 수기로 숫자를 타이핑하도록 되어있다. 이는 실제 여행에 있어서, 상당한 불편함을 초래하며, 숫자를 잘못 기입할 문제가 발생할 소지가 다분하다. 하지만, 카메라/사진 영수증 자동인식 기능이 정상적으로 동작한다면, 이용자는 순식간에 해외에서 영수증을 받아서 찍기만 하면, 손쉽게 예산, 지출내역을 관리할 수 있어 편의성이 매우 증대될 것으로 예측된다.

3. Glossary

This chapter specifies the terms used in this document.

Term	Description
Login	Sign in and log in are synonyms. Both mean "to open a session with an account that is already created"
Signup	To register; To create an account
Database	A backend term which represents a large collection of data
Database model	A type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized and manipulated. The most popular example of a database model is the relational model, which uses a table-based format
Api	An interface or communication protocol between a client and a server intended to simplify the building of client-side software. An API specification can take many forms, but often includes specifications for routines, data structures, object classes, variables, or remote calls
OCR	Optical character recognition or optical character reader is the electronic or mechanical conversion of images of typed,

Requirements Specification

	handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo.
Input	The act of entering data into a computer or data processing system
Object	In computer science, an object can be a variable, a data structure, a function or a method, and as such, is a value in memory referenced by an identifier. In relational database management, an object can be a table or column, or an association between data and a database entity(such as relating a person's age to a specific person)
Concurrent	A concurrent system is one where a computation can advance without waiting for all other computations to complete
Asynchronous(비동기)	The occurrence of events independent of the main program flow, and ways to deal with such events.

Table 1: Glossary

4. User Requirements Definition

4.1 Objective

사용자에게 시스템이 제공하는 서비스와 운용상의 제약사항, 비기능적 특성에 대해 설명한다.

요구사항은 Functional requirements와 non-functional requirements, external requirements로 분류하며, 사용자의 이해를 돕기 위해 각 요구사항에 대한 설명은 자연어를 사용하며 필요시 다이어그램을 활용한다.

4.2 Functional Requirements

A. Sign Up & Login

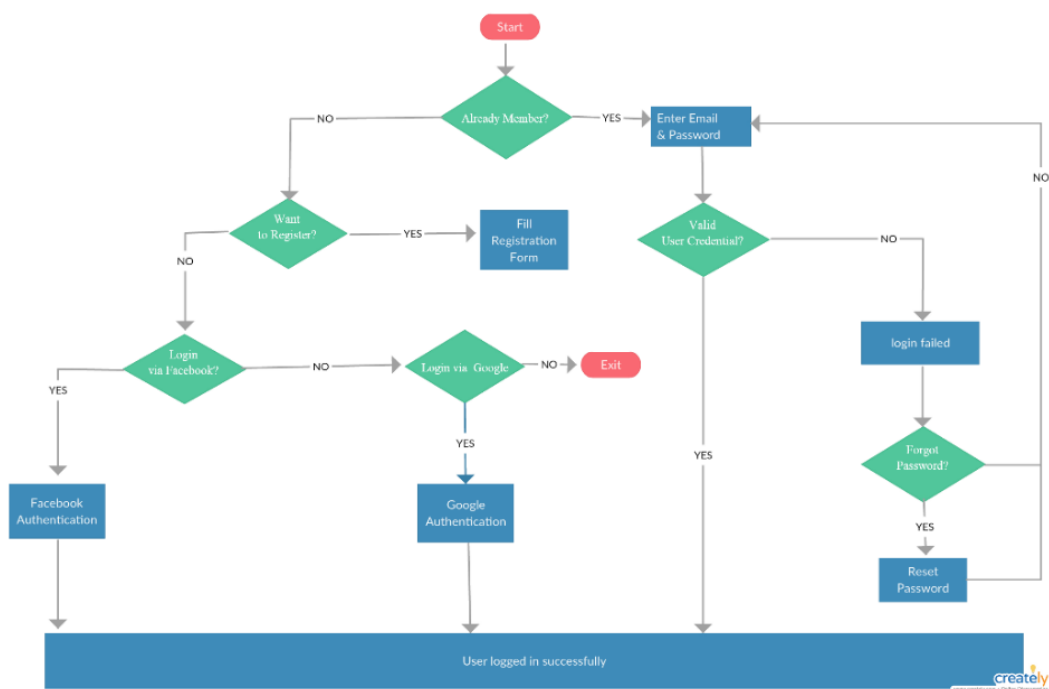


Diagram 1: Signup&Login Process

Requirements Specification

사용자를 구분하고 개인정보를 보호하기 위해 필요한 기능이다. Eco-travel를 사용하기 위해서는 ID와 PW가 등록되어 있어야 하며 첫 사용자는 Sign up을 통해 본인의 ID와 PW를 등록할 수 있다. Sign up을 이미 한 사용자는 ID와 PW를 입력해 로그인하고 시스템을 사용할 수 있다.

B. Travel Schedule Management

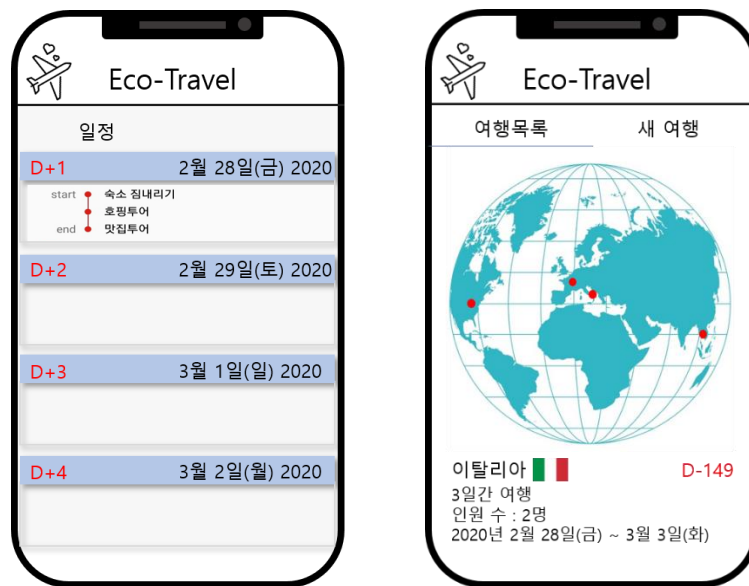


Figure 5: Travel Schedule Prototype

여행 일정을 관리하는 기능이다. 사용자는 여행을 하며 날마다 어디를 가서 무엇을 할지를 간단하게 기입할 수 있다. 사용자가 등록한 일정들은 일정 탭에서 하루단위 정리되어 시간 순으로 나열되며 같은 그룹 내의 모든 사람들은 이 일정을 공유하고 수정할 수 있다. 여행목록에서는 현재 진행중인 여행의 총 여행일과 남은 일 수, 인원 수와 시작일, 종료일이 요약되어 표시되며, 이미 시간이 지난 일정과 여행이 종료된 일정은 모두 기록되며 여행의 일원은 이 기록을 열람할 수 있다.

C. Budget Management

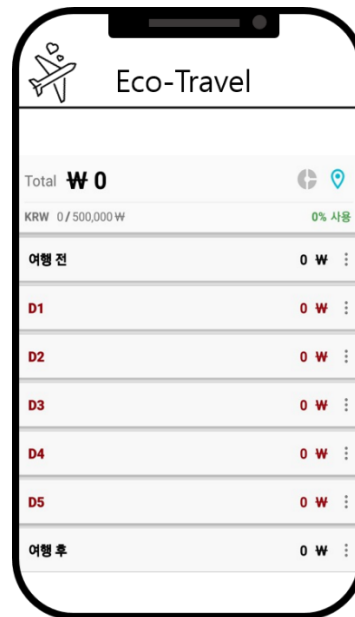


Figure 6: Budget Management Prototype

여행을 위해 준비한 예산과 여행 간의 지출을 기록하고 남은 돈을 계산해주는 기능이다. 사용자는 지출을 기입할 때 마다 전체 예산 중 얼마나 사용했는지 확인할 수 있다. 지출을 기입할 때의 위치를 기억해 나중에 해당 지출이 어디서 일어났는지 확인 가능하며 공식환율을 적용해 계산하는 기능을 활용할 수 있다. 지출을 한 시간을 활용해 여행 일정에도 하루 단위로 지출을 합산한 값을 표시한다.

D. Receipt Camera Recognition

Budget management에서 지출을 등록하는 방법 중 하나로 활용할 수 있는 기능이다. 스마트폰의 카메라 기능을 활용해 영수증을 화면에 보이면 영수증의 내용을 인식해 사용한 돈과 지출 내용을 자동으로 기입하여 예산 관리에 활용한다.

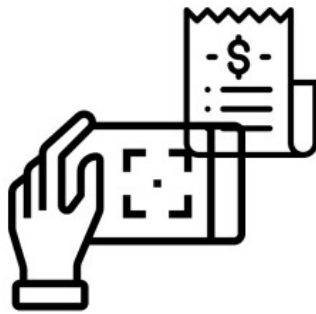


Figure 7: 영수증 카메라 자동인식(2)

E. Schedule Map

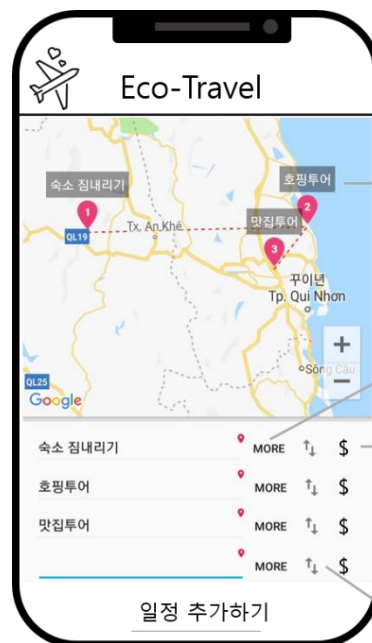


Figure 8: Schedule Map Prototype

Requirements Specification

여행지를 검색하는 기능이다. 여행 일정 등록 시 첫 목적지부터 각 일정을 보내게 될 장소를 검색할 수 있다. 구글맵을 연동하여 직접 지도를 스크롤하여 찾는 기능과 검색을 통해 찾는 기능이 있으며 검색은 자동 완성 기능을 제공한다.

F. Friends Invitation



Eco-Travel

여행목록 새 여행

1. 여행 시작일과 종료일을 선택하세요

2019 SEPTEMBER

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

2. 여행 목적지를 입력하세요

🔍

3. 같이 여행갈 친구를 초대하세요

Figure 9: Add Travel Prototype

여행을 같이 떠날 일행을 등록하는 기능이다. 이 어플리케이션 서비스에 회원가입이 되어 있는 유저를 검색하여 초대메시지를 보낼 수 있으며 메시지를 승낙하게 되면, 그 일행은 여행의 파티원으로 합류가 되어 여행일지를 읽고, 수정할 수 있는 권한이 주어지게 된다.

4.3 Non-functional Requirements

A. Product Requirements

1. Usability

사용자가 일정 및 예산을 관리할 때 직관적이며 필요한 정보를 모두 나타내어 쉽게 파악할 수 있어야 한다. 새로운 여행과 일정을 등록하는 것, 지출을 추가하고 예산을 관리하는 기능들이 각각 분리되어 사용자가 구분하여 사용할 수 있게 해야 한다.

2. Performance

여행 일정들과 연관된 지역, 해당 일정에서 사용한 지출과 남은 예산이 모두 정확하게 연결되어 사용자가 동시에 확인할 수 있어야 한다. 검색 기능은 사용자가 불편을 느끼지 않을 정도의 속도와 정확도가 보장되어야 하고, 자동 완성 기능은 사용자가 편의를 느낄 수 있도록 충분한 완성도를 가져야 한다. 영수증 인식 기능은 직접 지출을 기입하는 것보다 편리해야 하므로 충분한 속도와 정확도가 필요하고 최대한 다양한 형식의 영수증을 인식할 수 있어야 한다.

3. Dependability

여행 일정과 지출의 기록이 잘 보존되어야 한다. 일정과 지출은 1건을 단위로 저장되어 새로운 일정이나 지출의 기입이 통신이나 하드웨어의 문제 등으로 인해 실패할 경우 해당 건을 제외한 나머지 기록은 모두 보존되게 한다. 영수증 인식 기능의 경우 정확한 인식을 하지 못하고 정보를 기입하면 사용자가 시스템을 신뢰할 수 없기 때문에 만약 정확한 인식을 하지 못하는 경우에는 사용자에게 인식 실패를 알리고 해당 정보는 기입하지 않는다.

4. Security

Requirements Specification

사용자에 대한 모든 정보는 외부로부터의 보호가 보장되어야 한다. 사용자의 PW는 본인을 제외한 모든 사용자는 접근할 수 없으며, 여행 일정과 위치, 지출 정보는 해당 여행을 공유한 일행만이 열람할 수 있도록 한다.

B. Organizational Requirements

1. Environment

다양한 안드로이드 기반의 다양한 기기에서 사용 가능해야 한다. 여행을 하며 이용하게 되므로 데이터 접속이 제한적인 곳에서도 원활하게 사용할 수 있게 많은 양의 데이터 이동을 줄이고, 통신이 없더라도 이미 기록된 정보들을 사용자가 볼 수 있도록 해야 한다.

2. Operation

같은 그룹 내에서 정보를 공유해야 하므로 일행 사이의 통신이 가능해야 한다. 여행 일정과 지출이 이루어지는 위치를 파악해야 하므로 기기의 GPS기능을 사용한다. 여행을 같이할 일행을 초대하는 메시지를 보내기 위한 카카오톡, 여행 일정을 보내고 여행을 할 장소를 검색하기 위해 구글맵과의 연동이 필요하다.

3. Development

영수증 인식 기능을 위해 머신러닝을 통한 이미지 인식이 필요하므로 이를 위해 TensorFlow와 OpenCV를 사용한다. 백엔드 서버는 Node.js를, 프론트엔드는 안드로이드 기반으로 Glide와 Retrofit2의 라이브러리를 활용한다.

C. External Requirements

1. Regularity

시스템은 사용자의 Sign up부터 사용과정에서 개인정보를 수집하게 된다. 사용자의 개

Requirements Specification

인정보는 Sign up에서 제공되는 이용약관에 대한 동의를 얻고 수집을 시작하며 약관에 따라 정보를 관리한다. 외부에 정보를 제공하게 될 경우 사용자의 동의를 우선적으로 받아야 하며, 약관에 어긋나는 개인정보에 대한 외부의 접근은 발생하지 않도록 한다.

2. Safety, Security

사용자의 개인정보를 가지고 있는 서버는 외부로부터의 접근으로부터 정보를 보호해야 한다. 또한 보안에 문제가 생기더라도 피해를 최소화하기 위해 시스템에 필요한 정보만을 수집한다. 영수증 인식에 있어서 잘못된 정보를 기입하지 않도록 한다.

5. System Architecture

이 챕터에서는 시스템의 개괄적인 Architecture를 기술한다. 시스템의 전체적인 구조와 각 서브시스템의 구성, 서브시스템간 관계를 대략적으로 설명하며, 각 구조는 다이어그램을 첨부하여 이해를 돕는다.

5.1 Frontend Architecture

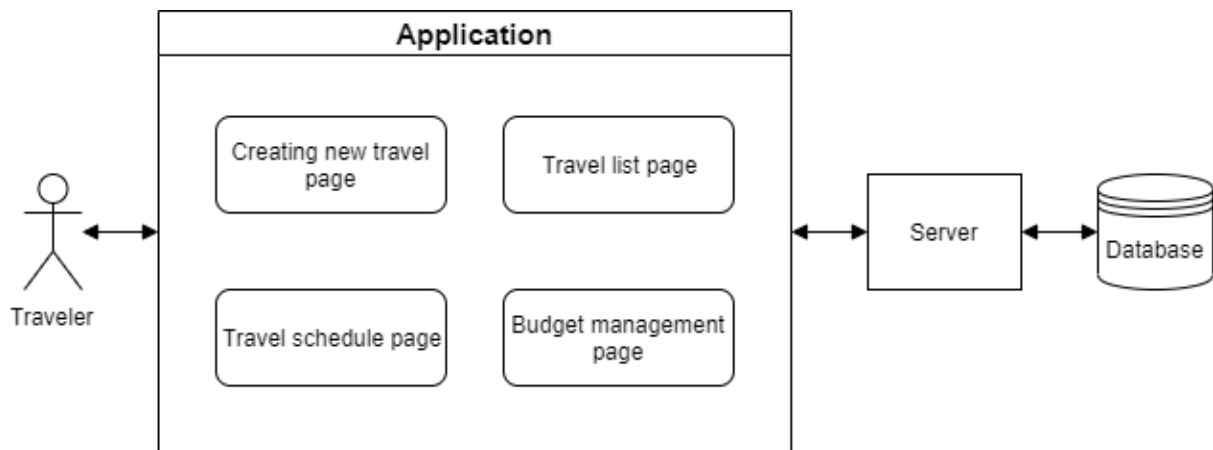


Diagram 2: Frontend Architecture

전체 프로젝트 구조는 크게 앱, 서버, 데이터베이스로 이루어져 있다. 그 중 유저는 Frontend 인 앱으로 서비스를 이용한다. 앱은 새 여행 등록 페이지, 여행 리스트 확인 페이지, 여행 일정 관리 페이지, 예산 관리 페이지로 구성된다. 각 페이지는 서버를 통해 데이터를 가져와 보여주고 생성되거나 수정된 데이터들을 서버로 보내 데이터베이스에 저장되도록 한다.

5.2 Socket Communication System

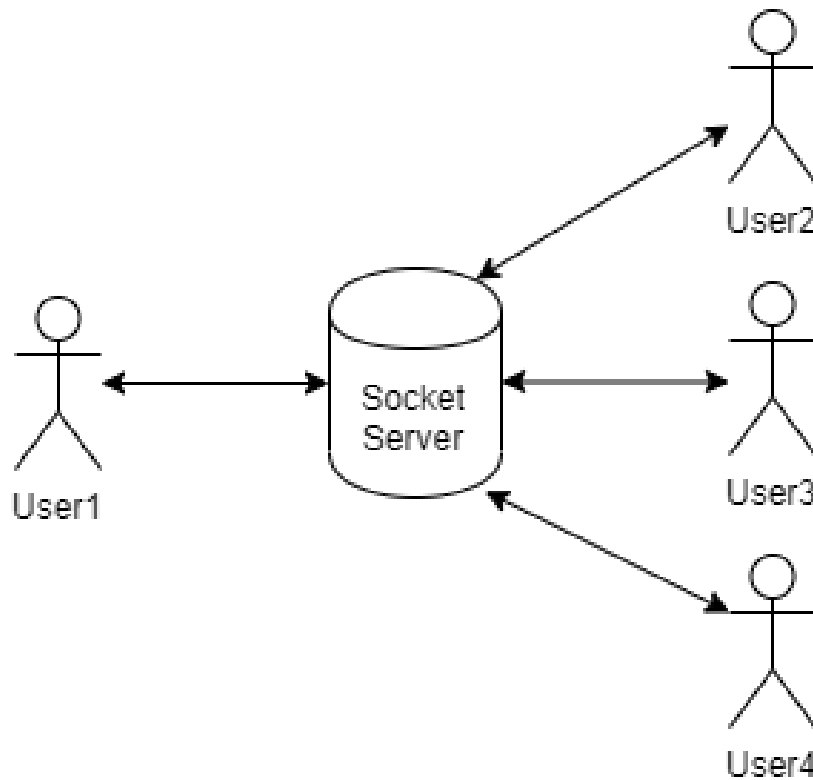


Diagram 3: Socket Communication

또한, 팀을 이룬 여행동반자들은 하나의 namespace로 묶이게 되며, 접속을 하거나, 일정, 예산 지출 내역 작성 혹은 수정을 하는 이벤트 발생시에 서버는 이를 수신하고 같은 namespace 내에 있는 모든 구성원들에게 멀티캐스팅을 하여 이벤트를 푸시한다. 이렇게 함으로써, 한 사용자가 일정을 생성하거나 수정하는 즉시 다른 유저에게도 반영이 되는 실시간 동기화가 가능해진다.

5.3 Backend Architecture

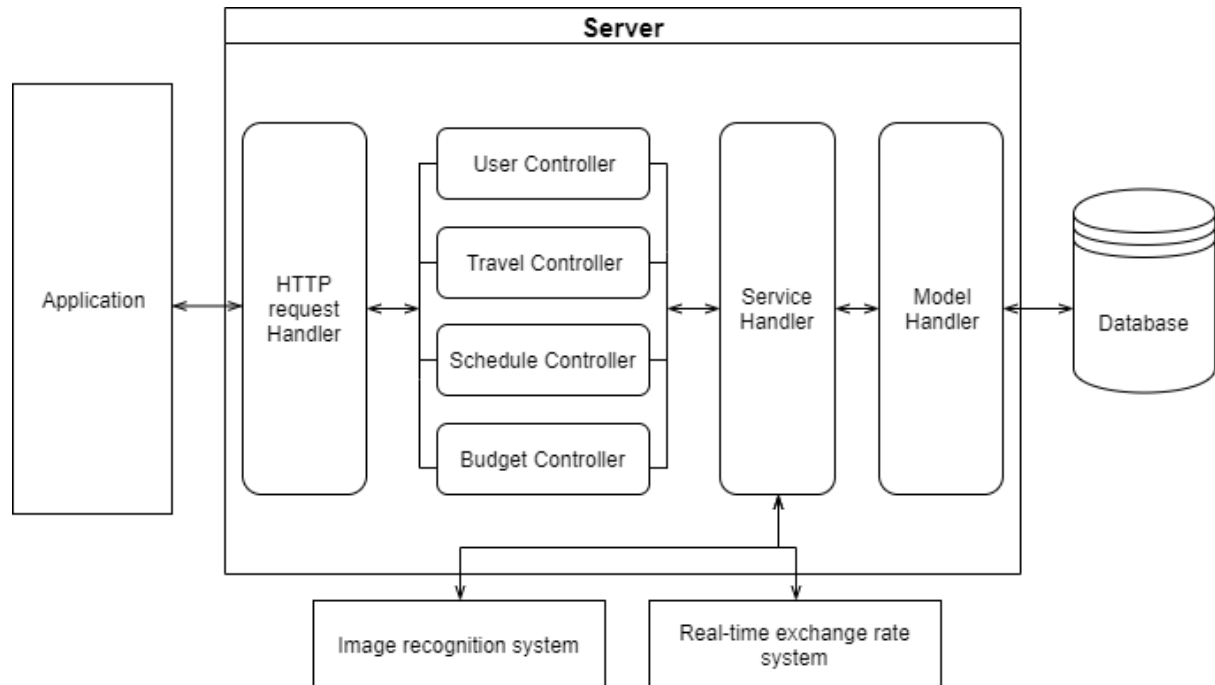


Diagram 4: Backend Architecture

데이터를 처리하는 서버의 구조는 위와 같다. 앱에서 받은 HTTP 요청을 해당하는 컨트롤러에 보내 처리한다. 필요한 경우 서비스 핸들러를 통해 이미지 인식이나 환율 계산을 마친 후 모델 핸들러를 거친다. GET/POST 등 요청의 종류에 따라 데이터베이스에 저장되거나 저장된 데이터를 앱으로 보내 사용자가 읽을 수 있도록 한다.

5.4 Image Recognition System

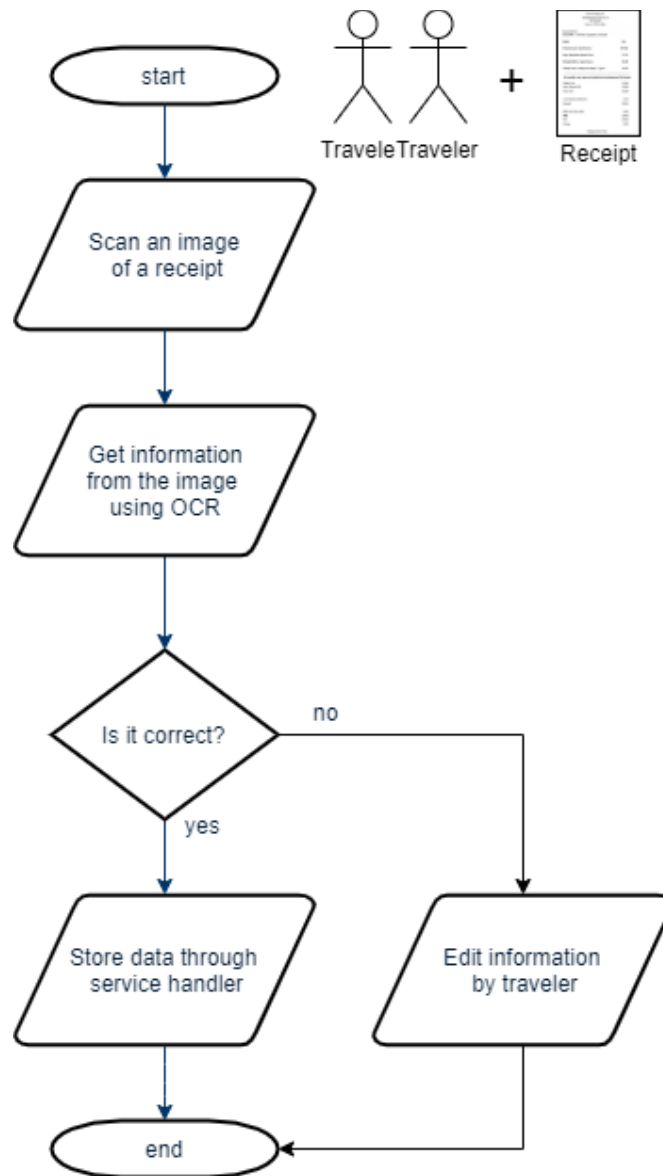


Diagram 5: Flowchart of Getting Receipt Info

본 앱은 스마트폰의 카메라 기능과 OCR(Optical character recognition) 기술을 활용한 이미지 인식으로 영수증의 정보를 자동으로 앱에 입력할 수 있다. 인식 후 입력된 정보를 확인한 후 실제 정보와 다르게 인식된 경우 사용자가 수정할 수 있도록 하여 정확한 정보를 저장할 수 있도록 한다.

5.5 Real-time Exchange Rate System

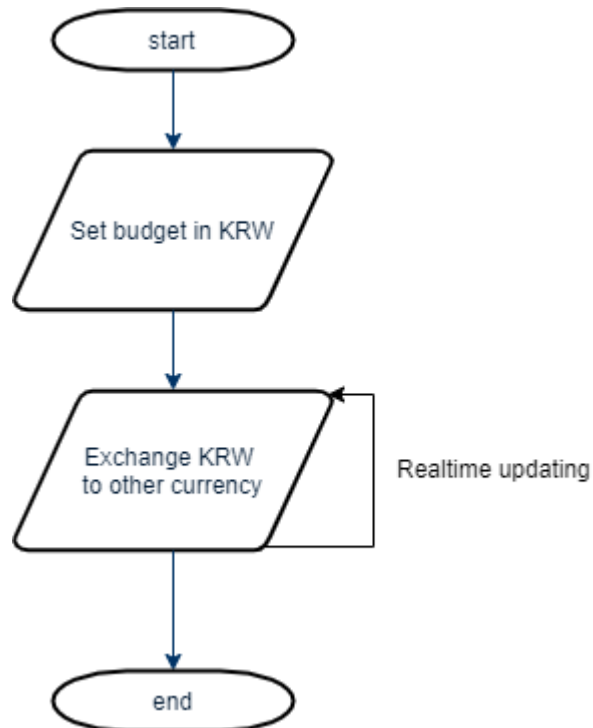


Diagram 6: Exchange Rate System

본 앱에서는 예산 관리를 원화뿐 아니라 여행 간 현지 통화로도 가능하게 하기 위해 실시간 환율을 반영해 환전할 수 있도록 하였다. 사용자의 편의를 위해 고정 환율로도 환전 가능하도록 한다.

6. System Requirements Specification

6.1 Objective

이 챕터에서는 유저 요구사항 명세서에서 정의된 요구사항을 바탕으로 세부적인 Functional requirements와 Non-functional requirements를 명시한다. 각 requirements는 Name, description, inputs, source, outputs, destination, action, requirements, pre-condition, post-condition, other activities, side-effects를 포함해 최대한 상세히 기술한다.

6.2 Functional Requirements

A. 로그인

Name	Def. Login
Description	사용자 편의성을 위하여 custom으로 제작한 로그인 양식외에 카카오, 페이스북, 구글 로그인 api를 적용한다. 로그인 정보 저장하기를 통하여 이용자 편의를 도모한다.
Inputs	id, password
Source(Input)	유저 인풋
Outputs	User model로부터 인증된 회원 정보 객체 반환
Destination(Output)	시작화면(회원가입/로그인) * 단, 로그인 저장에 되어있다면, 회원가입/로그인 화면은 생략된다.
Action	아이디와 비밀번호를 입력하면 user 테이블에 쿼리를 보내 아이디와 비밀번호가 일치하는 user를 찾는다. user 객체 반환과 함께 여행목록 전체리스트 화면으로 redirect한다.
Requirements	만약 없는 아이디거나, 비밀번호가 맞지 않다면, "잘못된 아이디거나 비밀번호입니다."라고 인풋 form table 아래에 출력한다.
Pre-condition	유저는 회원가입을 먼저 진행한 후, 로그인을 할 수 있다. 유저는 non-authorized tap만 이용할 수 있다.
Post-condition	로그인된 유저는 모든 authorized tap에 접근이 가능하다.

Table 2: Login Function

Requirements Specification

B. 회원가입

Name	Def. Join
Description	사용자 편의성을 위하여 custom으로 제작한 회원가입 양식외에 카카오, 페이스북, 구글 로그인 api를 적용한다.
Inputs	Name, id, password, nickname
Source(Input)	유저 인풋
Outputs	Insert into User model
Destination(Output)	시작화면(회원가입/로그인) * 단, 로그인 저장에 되어있다면, 회원가입 화면은 생략된다.
Action	아이디와 닉네임 입력란을 작성함과 동시에 비동기적으로 아이디 필드값과 닉네임 필드값을 user 테이블에서 대조하여 중복체크를 하여 경고메시지를 출력한다. Requirement에 맞게끔 입력란이 모두 작성되었으면, user는 회원가입 버튼을 누른다. 회원가입 버튼을 누름과 동시에, 비밀번호 필드값은 암호화되고, 다른 필드값과 함께 user 테이블에 저장되고, 로그인과정까지 자동적으로 진행된다.
Requirements	아이디, 비밀번호, 닉네임은 모두 unique 값이어야 한다. 아이디 : 20자 이내 string 문자열 비밀번호 : 8~16자 특수문자, 알파벳이 포함된 string 문자열 닉네임 : 20자 이내 string 문자열
Pre-condition	유저는 non-authorized tap만 이용할 수 있다.
Post-condition	회원가입과 동시에 로그인되며, 모든 authorized tap에 접근이 가능하다.

Table 3: Signup Function

C. 새 여행 추가하기

Name	Def. Create Travel
Description	로그인된 유저가 새로운 여행을 추가한다.
Inputs	여행 시작일, 종료일(date_start, date_end) => 달력에서 drag로 선택 여행 국가(nation_id) 여행 도시(city_id) 여행 일행(user_id)
Source(Input)	버튼 클릭
Outputs	먼저, 생성된 travel 객체가 insert into travel model. 이후, User와 Travel model은 M대 N관계이므로, mapping table로 다대다 관계를 처리한다. Mapping table에 여행 일행을 포함한 모든 user_id와 형성된 travel객체의 travel_id들이 mapping된다. 또한, 여행 일행이 초대를 수락함과 동시에, mapping table에 기록된다.
Destination(Output)	새 여행 추가하기 화면
Action	로그인된 유저가 새 여행 추가하기 버튼을 클릭하면, 여행목록 탭에서 자신이 생성한 여행이 여행목록에 추가가 되었음을 확인할 수 있다. 또한, 여행 상세보기를 통하여 특정 여행과 관련된 정보(날짜, 친구추가, 국가)를 수정할 수 있다. 한편, user들에게 초대 메시지가 전송된다.

Requirements Specification

Requirements	여행 시작일과 종료일은 필수 입력값이다. 여행 목적지는 필수 입력값이다. 여행 동행자는 필수 입력값이 아니다.(1인 여행자 고려)
Pre-condition	홈화면(= 여행 목록 리스트 화면)에 여행 목록이 존재X
Post-condition	홈화면(= 여행 목록 리스트 화면)에 추가한 여행이 목록에 표시됨

Table 4: Add New Travel Function

D. 여행 목적지(국가, 도시) 검색

Name	Def. Find_destination
Description	검색한 국가를 보여준다.
Inputs	검색할 국가(nation_id), 검색할 도시(city_id)
Source(Input)	유저 인풋
Outputs	검색string과 일치하는 국가, 도시를 인풋 form바로 밑에 슬라이드 형식으로 보여준다. 유저가 국가, 도시를 선택하면 nation_id, city_id가 서버로 전송될 인풋 form attribute에 추가된다.
Destination(Output)	새 여행 추가하기 화면
Action	유저가 인풋 form에 여행 갈 국가명, 도시명을 typing한다. 유저가 원하는 국가와 도시가 슬라이드화면에 나타나면 이를 터치하면 완료.
Requirements	국가, 도시 리스트를 데이터베이스에 미리 저장해놓는다.
Pre-condition	여행 추가하기 화면 - 1. 날짜 입력(터치)에 커서 위치
Post-condition	여행 추가하기 화면 - 3. 같이 여행 갈 친구 초대에 커서 위치

Table 5: Search Destination Function

E. 여행 같이 갈 친구 검색하기

Name	Def. Invite_friends
Description	여행 같이 갈 친구들을 검색한다.
Inputs	유저명(String) or 닉네임(Nickname)
Source(Input)	유저 인풋
Outputs	User model에서 유저가 입력한 닉네임 혹은 이름과 일치하는 user를 찾아서 user 객체를 가져온다. 유저가 초대하기를 누르면, 선택된 user들의 user_id들을 인풋 form attribute에 추가한다.
Destination(Output)	새 여행 추가하기 화면
Action	유저가 인풋 form에 여행 같이 갈 친구 닉네임 혹은 이름을 검색한다. 검색된 친구가 화면에 나타나면, 유저가 이를 선택한다. 추가된 유저는 인풋 폼 하단에 표시된다. 만약 잘못된 유저를 추가했을 경우, 표시된 유저를 삭제할 수 있다. 추가적으로 검색된 친구가 없다면, "닉네임 혹은 이름을 확인해주십시오"라는 메시지를 인풋 폼 하단에 출력한다.
Requirements	검색하려는 친구는 이미 회원가입이 되어 user model에 저장된 상태여야 한다.

Requirements Specification

Pre-condition	여행 추가하기 화면 - 2. 여행 목적지 입력에 커서 위치
Post-condition	여행 추가하기 화면 - 마지막 항목인 여행친구 검색이 끝나면, 필수항목 입력값들을 체크하여 모든 필수항목이 입력되었으면, 여행 추가하기 화면이 어두워지고, 화면 위에 "여행추가 완료" or "돌아가기" 버튼이 보여진다. 필수항목이 입력되지 않았으면, 입력되지 않은 항목들을 표시해준다.

Table 6: Search Friends Function

F. 여행목록 전체리스트 보여주기

Name	Def. Show_TravelList
Description	로그인된 사용자가 현재시점까지 생성한 모든 여행 오브젝트들을 서버로부터 받아와서 그 중 중요한 데이터들을 리스트 형식으로 구분하여 유저에게 보여준다.
Inputs	User_id, travel_id
Source(Input)	버튼 클릭
Outputs	Mapping table에서 user_id와 travel_id와 일치하는 travel 객체를 찾아서 반환.
Destination(Output)	로그인된 유저의 홈화면
Action	로그인된 유저가 유저 홈화면에서 여행목록 버튼을 누르면, travel 테이블에서 user_id를 foreign key로 가지고 있는 travel 오브젝트들의 필드값 중에 위에서 명시한 outputs에 해당하는 필드값들을 불러온다. 각 여행지별로, 불러온 필드값들을 양식에 맞게끔 정렬하여 리스트형태로 보여준다. 만약, 유저가 생성한 travel 오브젝트가 하나도 없는 경우, 여행지를 추천하는 글귀와 함께 여행지 사진이 랜덤으로 제시된다.
Requirements	<p>*각각의 여행리스트안에는 아래의 내용들이 보여져야 한다.</p> <ul style="list-style-type: none"> - 지구본상에서의 여행지 위치 - 여행지 국가, 도시명(nation_id, city_id) - 여행 D-day(만약 이미 완료한 여행이라면 D+# 형식으로 보여준다.) - 여행 일정, 날짜 - 같이 간 일행 이름, 수 <p>유저가 새로운 여행 오브젝트를 만들었을 경우 => 여행목록리스트O 유저가 새로운 여행 오브젝트를 만들지 않았을 경우 => 여행지추천화면</p>
Pre-condition	홈화면(= 여행 목록 리스트 화면)
Post-condition	홈화면(= 여행 목록 리스트 화면)
Other activities	여행 오브젝트에 묶여 있는 다른 여행 참가자가 여행과 관련한 정보를 수정하는 이벤트가 발생하는 경우, 이는 서버에 저장되고, 서버는 그 특정한 여행 오브젝트에 속해 있는 모든 구성원의 정보를 갱신하여 보여준다.
Side-effects	서버로부터 정보를 온전히 수신받지 못하였을 경우, 에러가 발생할 수 있다.

Table 7: Show Travel List Function

Requirements Specification

G. 여행 상세보기

Name	Def. Show_Travel
Description	유저가 선택한 특정 여행 오브젝트의 모든 정보를 보여준다.
Inputs	User_id, travel_id
Source(Input)	여행리스트 클릭
Outputs	Mapping table에서 user_id와 travel_id와 일치하는 travel 객체를 찾아서 반환.
Destination(Output)	여행 상세보기 화면
Action	여행목록 리스트에서 특정 여행을 클릭하면, travel 테이블에서 user_id, travel_id와 일치하는 여행 오브젝트를 불러와서, 여행 상세보기 화면에 보여준다.
Requirements	일정관리 탭과 예산관리 탭 일정 탭을 디폴트화면으로 지정한다. * D. 여행일정 보여주기와 outputs이 같으므로 이를 참고한다.
Pre-condition	여행목록 탭에 여행 오브젝트들이 있어야 가능하다. 여행목록 보여주기 화면
Post-condition	여행 상세보기 화면
Other activities	여행 오브젝트에 묶여 있는 다른 여행 참가자가 여행과 관련한 정보를 수정하는 이벤트가 발생하는 경우, 이는 서버에 저장되고, 서버는 그 특정한 여행 오브젝트에 속해 있는 모든 구성원의 정보를 갱신하여 보여준다.
Side-effects	서버로부터 정보를 온전히 수신받지 못하였을 경우, 에러가 발생할 수 있다.

Table 8: Show Travel Detail Function

H. 여행일정 보여주기

Name	Def. Show_schedule
Description	특정 여행 오브젝트의 상세 일정을 보여준다.
Inputs	User_id, travel_id
Source(Input)	유저 클릭
Outputs	여행날짜별로 탭을 만들어 리스트 형식으로 보여준다. 각각의 탭안에서 D-#, 여행날짜 및 여행일정리스트, 하루예산을 보여준다.
Destination(Output)	여행 일정 보여주기 화면
Action	travel 테이블에서 user_id, travel_id와 일치하는 여행 오브젝트를 불러와서, 여행 일정테이블을 여행날짜별로 보여준다.
Requirements	여행 상세보기를 눌렀을 경우에만 일정관리탭에 접근이 가능하다.
Other activities	여행 오브젝트에 묶여 있는 다른 여행 참가자가 여행과 관련한 정보를 수정하는 이벤트가 발생하는 경우, 이는 서버에 저장되고, 서버는 그 특정한 여행 오브젝트에 속해 있는 모든 구성원의 정보를 갱신하여 보여준다.
Side-effects	서버로부터 정보를 온전히 수신받지 못하였을 경우, 에러가 발생할 수 있다.

Table 9: Show Travel Schedule Function

Requirements Specification

I. 여행일정 추가(수정)하기

Name	Def. Create_schedule
Description	유저가 선택한 특정 여행 오브젝트의 특정 여행날짜의 일정을 추가한다.
Inputs	여행일정 내용(String) 일정 장소(Google-Map api) 일정 순서 변경(위, 아래) 각 일정마다 예산(Float)
Source(Input)	유저 인풋
Outputs	Insert into Schedule table Schedule table에 foreign key로 travel_id 기록 (Travel model : Schedule model = 1 : N)
Destination(Output)	여행 추가하기 화면
Action	여행목록 리스트에서 특정 여행을 클릭하면, travel 테이블에서 user_id, travel_id와 일치하는 여행 오브젝트를 불러와서, 여행 상세보기 화면에 보여준다. 따로 수정하기 버튼은 있지 않으며, 완성된 일정을 클릭하면 일정을 수정할 수 있다.
Requirements	일정 보여주기 화면에서 일정이 추가됨 일정 순서에 맞게끔 여행 동선이 표시됨
Pre-condition	여행 일정 보여주기 화면에 일정 없음
Post-condition	여행 일정 보여주기 화면에 일정 추가됨
Other activities	여행 오브젝트에 묶여 있는 다른 여행 참가자가 일정과 관련한 정보를 수정하는 이벤트가 발생하는 경우, 이는 서버에 저장되고, 서버는 그 특정한 여행 오브젝트에 속해 있는 모든 구성원의 정보를 갱신하여 보여준다.
Side-effects	Reader-writer concurrent 문제가 발생할 수 있기 때문에, writer는 언제나 1명으로 제한한다.

Table 10: Add(Edit) Travel Schedule Function

J. 예산관리 보여주기

Name	Def. Show_budget
Description	유저가 선택한 특정 여행의 모든 일정에 대한 예산을 확인할 수 있다.
Inputs	Travel_id
Source(Input)	유저 클릭
Outputs	Select * from schedule Where travel_id = # Schedule 객체
Destination(Output)	여행 예산관리 화면
Action	여행 예산관리 탭을 클릭하면, travel_id가 일치하는 모든 schedule 테이블의 item들에서 schedule_id를 foreign key로 가지고 있는 budget 테이블의 item 데이터들을 모두 가져온다. 그리고 위에 Output에 맞게끔 데이터의 처리를 거쳐서 유저에게 보여준다. 일정관리에서도 예산을 수정할 수 있지만, 예산관리 화면에서도 예산을 수정할 수 있다.
Requirements	여행 총 경비

Requirements Specification

	날짜별 경비 날짜별 경비별 탭을 리스트로 보여준다. 총 예산에서 남은 경비, 사용비율 일정과 일정에 따른 상세 경비정보
Pre-condition	여행 일정관리 화면
Post-condition	여행 예산관리 화면
Other activities	여행 오브젝트에 묶여 있는 다른 여행 참가자가 일정과 관련한 예산정보를 수정하는 이벤트가 발생하는 경우, 이는 서버에 저장되고, 서버는 그 특정한 여행 오브젝트에 속해 있는 모든 구성원의 정보를 갱신하여 보여준다.
Side-effects	Reader-writer concurrent 문제가 발생할 수 있기 때문에, writer는 언제나 1명으로 제한한다.

Table 11: Show Budget Management Function

K. Show pay-map

Name	Def. Show_payMap
Description	유저의 여행일정 동선과 함께 예산지출 장소와 가격이 지도에 표시된다.
Inputs	Budget place와 그에 따른 budget
Source(Input)	유저 클릭
Outputs	Google map
Destination(Output)	여행 예산관리 화면
Action	payMap 아이콘을 클릭하면, 일정에 따른 budget item들을 budget 테이블에서 받아와서, budget place 필드값이 존재하는 경우, budget place를 google-map api에 넘겨서 장소 데이터를 받아오고, google-map에 이를 picker로 지정하고, 지출내역과 함께 표시한다.
Requirements	여행일정 동선(날짜별로 다른 색으로 표시) 예산지출 장소에 picker가 찍히고, 예산지출(W)가격 표시 *예산지출 장소가 없는 경우, 지도에 표시되지 않는다. *예산지출 장소를 특정할 수 없는 경우, 지도에 표시하지 않는다.
Other activities	여행 오브젝트에 묶여 있는 다른 여행 참가자가 일정과 관련한 예산정보를 수정하는 이벤트가 발생하는 경우, 유저가 pay-map을 이미 동작중인 경우, 이벤트 수신후, google-api에 다시금 데이터를 전송하여 맵정보를 갱신토록 한다.
Side-effects	맵데이터를 갱신시키는 과정에서 화면끊김이 발생할 수 있다.

Table 12: Show Pay-map Function

L. 영수증 자동인식

Name	Def. AutoRecognize_bill
Description	영수증의 총계값을 인식하여 읽어와서, 사진을 찍기만해도 가격이 자동적으로 입력된다.
Inputs	영수증 사진

Requirements Specification

Source(Input)	카메라, 앨범
Outputs	영수증의 총계값(Total Price) string으로 인식한 후, 이를 float값으로 변형
Destination(Output)	여행 예산관리 화면
Action	유저가 영수증 자동인식 카메라 버튼을 누르면, 카메라나 앨범에서 선택하는 화면으로 전환된다. 카메라를 선택하여 영수증을 촬영하거나 앨범에서 영수증을 선택하여 영수증의 인식이 정상적으로 완료되면, 카메라, 앨범이 종료되고, 여행 예산관리 화면으로 돌아와 해당 예산항목에 값이 자동적으로 기입된다. 만약 정상적으로 인식되지 않았다면, 정상적으로 인식되지 않았다는 메시지를 출력하고, 값이 자동으로 입력되지 않으며, 다시 찍기 or 수동기입을 선택하라는 문구가 보여진다.
Requirements	사진이 흐릿하거나 구겨져있다면, 인식이 안될 수 있기 때문에 선명하고 구겨지지 않은 상태의 영수증의 정면 사진을 요구함.
Pre-condition	여행 예산관리 화면
Post-condition	여행 예산관리 화면 - 해당 예산항목에 값이 자동적으로 기입됨

Table 13: Receipt Recognition Function

6.3 Non-functional Requirements

A. Product Requirements

1. Usability Requirements

본 시스템은 자국민의 해외에서 모바일 플랫폼을 통한 접속을 전제한다. 직관적이고 이해하기 쉬운 UI를 지향한다. 가장 중요한 것은 사용자가 기록을 하고, 관리를 하고 싶게끔 현대적이고 모던한 UI를 갖추는 것이다. 또한, 유저의 인풋을 많이 받는 서비스이므로, UI/UX에 있어서 사용자 편의, 관습을 해치지 않게 유념하여야 한다.

2. Performance Requirements

유저와 서버간의 통신과 서버와 멀티유저들간의 멀티캐스팅이 원활하게 이루어져야 하며, 특정 여행그룹에서 유저이벤트가 발생하였을 경우, 서버에서 이를 확인하고, 다시 멀티캐스팅을 통하여 모든 여행그룹이 이벤트를 수신하여 화면에 반영되는 데 걸리는 시간은 3sec이내여야 한다. Google map api를 사용함에 있어서 기록한 여행일정 동선과 다른 지도를 보여주어선 안된다. 모든 화면의 응답 속도는 3sec를 넘기지 않는다. 영수증 자동 인식에 걸리는 시간은 1sec이내여야 한다.

3. Dependability Requirements

회원가입 정보는 유출되어서는 안되며, 다수의 데이터 모델을 사용함에 있어서 데이터 베이스는 안전성을 보장해야한다. 또한, 영수증 카메라 자동인식 기능의 경우, 영수증에서의 숫자와 다른 숫자가 인식되어서는 절대로 안된다. 가장 중요한 것은 해외여행시에는 통신이 불안정할 수 있기 때문에, 서버로부터 데이터가 갱신되지 않는다고 하더라도, 예러로 인하여 메인 프로그램이 Knock-out되면 안된다.

Requirements Specification

4. Security Requirements

유저들의 사적인 정보가 저장되는 만큼, 데이터베이스는 보안공격에도 안전해야 한다. 또한, 유저들의 비밀번호는 암호화되어 저장되어야 한다. 여행 일행이 아님에도 불구하고, 다른 유저들의 여행일지를 볼 수 있어서는 안된다.

B. Organizational Requirements

1. Operational Requirements

React-native, Android-studio, node-js, AWS, mysql를 이용하여 개발한다. 개발언어는 javascript입니다. OpenCV, OCR모델을 활용하여 사진 속 글자를 인식하여 검출한다.

2. Development Requirements

팀원 전원이 안드로이드 스튜디오를 사용한적이 없으며, react-native와 관련한 기술을 배워본 적이 없기 때문에, 최대한 오픈소스 component를 활용하여 개발을 하도록 한다. 또한, 개발 프로세스는 react-native(프론트)와 서버구축, OCR(딥러닝) 세 파트로 나누어서 parallel development를 진행한다.

C. External Requirements

1. Regulatory Requirements

사용자가 자발적으로 만든 여행일지 정보의 소유권은 사용자 본인에게 있다. 따라서, 사용자들의 허가없이, 본 시스템 관리자는 어떠한 용도로도 사용자들의 여행일지 정보를 가공, 외부에 제공할 수 없다. 정보의 공개 범위는 사용자 본인과 관리자까지이며, 비밀번호의 경우 사용자 본인만이 알 수 있다. 만약 사용자가 자신의 정보를 삭제하기를 원하는 경우, 본 시스템 관리자에게 문의하여 삭제를 요청할 수 있으나 향후 서비스 이용이 제한된다.

2. Safety/Security Requirements

개인정보보호법에 따라 고객 개인정보를 안전하게 보관하고 보호하여야 한다.

6.4 Scenario Examples

A. Add New Travel Schedule Scenario

1. Initial assumption

사용자는 로그인을 통해 홈화면에서 일정관리화면으로 리다이렉션되었다. 사용자는 어떤 국가, 도시, 그리고 관광지로 여행갈지, 또한 누구랑 여행을 같이 갈지 이미 정해놓은 상황에서 여행 일정을 생성하고자 한다. 사용자는 물론 같이 여행을 가려는 동행자 모두 본 시스템에 회원가입이 되어 있다고 가정한다. 서버와 통신해야 하기 때문에, 기기에 인터넷이 연결되어 있어야 한다.

2. Normal flow of events

일정관리화면의 상단에는 ‘여행목록’과 ‘새 여행’의 두가지 탭이 표시되어 있고, 사용자는 ‘여행일정 생성하기’ 버튼을 누른다. 새로운 여행일정 생성화면으로 리다이렉션되며, 여행 시작일, 종료일을 달력위젯에서 터치앤 드래그 방식으로 설정한다. 여행 날짜 선택이 완료되면, 여행 국가와 도시를 검색하여 입력하게 된다. 검색을 치면, 비동기적으로 검색명과 일치하는 여행국가와 도시가 슬라이드 형식으로 표시되고, 슬라이드에 검색되서 나온 여행국가와 도시를 찾아서 누르게 되면, 입력이 완료된다. 마지막으로, 유저는 같이 여행을 갈 친구를 검색한다. 이때, 유저는 닉네임과 이름을 통해 검색을 할 수 있으며, 국가, 도시 검색 때와 마찬가지로 슬라이드 형식으로 검색결과가 표시된다. 유저가 여행 갈 친구 선택을 완료하고, 최종적으로 여행 일지 생성 버튼을 누른다.

Requirements Specification

3. What can go wrong

필수입력란인 여행시작일, 종료일 선택과 여행 목적지를 선택하지 않고 여행 일지 생성 버튼을 누르면, 필수 입력란을 기입하지 않았다는 메시지가 출력되고, 기입되지 않은 필수 입력란은 빨간색으로 표시된다. 또한, 초대하려고 하는 친구가 만약 시스템에 등록되지 않았다면, 일치하는 검색결과가 없으며, 회원가입된 유저인지 확인해주세요라고 표시한다.

4. System state on completion

여행목록에 생성한 여행이 탭으로 표시되고, 이를 누르면, 세부 일정을 입력하고, 예산, 지출 내역을 관리할 수 있게 된다.

B. Camera Receipt Recognition Scenario

1. Initial assumption

사용자는 이 시스템의 기기에 대한 카메라/앨범에 대한 권한을 승인한 상태여야 한다. 사용자는 하나의 새로운 여행을 생성한 상태이다.

2. Normal flow of events

사용자는 로그인 후, 홈화면에서 표시된 여행목록들 중, 원하는 여행을 선택하여 누른다. 상단에 '일정관리'와 'budget 관리'가 표시되고, 사용자는 '일정 관리' 버튼을 누른다. 사용자는 특정 일정을 생성하여 예산, 지출내역을 입력할 수 있다. 이때, '영수증 자동인식' 버튼을 누르게 되면, 사용자에게 카메라/앨범에 대한 최초 권한을 묻게 되고, 사용자가 이를 수락하게 되면, 카메라/앨범 둘 중에서 선택하게 된다. 카메라를 선택하여 영수증을 촬영하거나, 앨범에서 영수증 사진을 선택한다.

Requirements Specification

3. What can go wrong

카메라/앨범 영수증이 화질이 좋지 않거나, 정면에서 찍은 사진이 아니거나, 훼손이 되어있는 상태라면, 자동인식 기능이 제대로 작동하지 못할 확률이 높다. 이 때는, 인식이 불가능하다는 오류 메시지를 출력하고 사용자에게 재촬영을 요구하거나 값을 직접 기입하도록 알림을 준다.

4. System state on completion

영수증의 총계부분의 숫자값이 인식되어 자동으로 일정 지출내역관리 입력폼에 기입된다.

7. System Models

이 챕터에서는 각 시스템 컴포넌트와 전체 시스템, 그리고 시스템을 둘러싼 환경 사이의 관계를 여러 Diagram을 통해 기술한다.

7.1 Context Models

A. Context Diagram

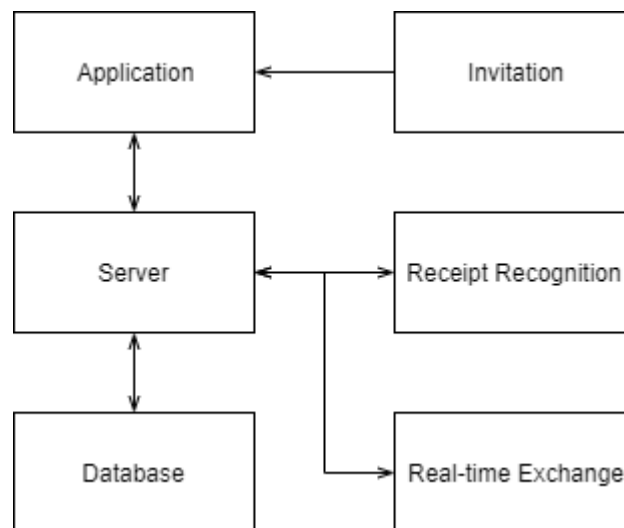


Diagram 7: Overall Context Diagram

Requirements Specification

B. Process Diagram

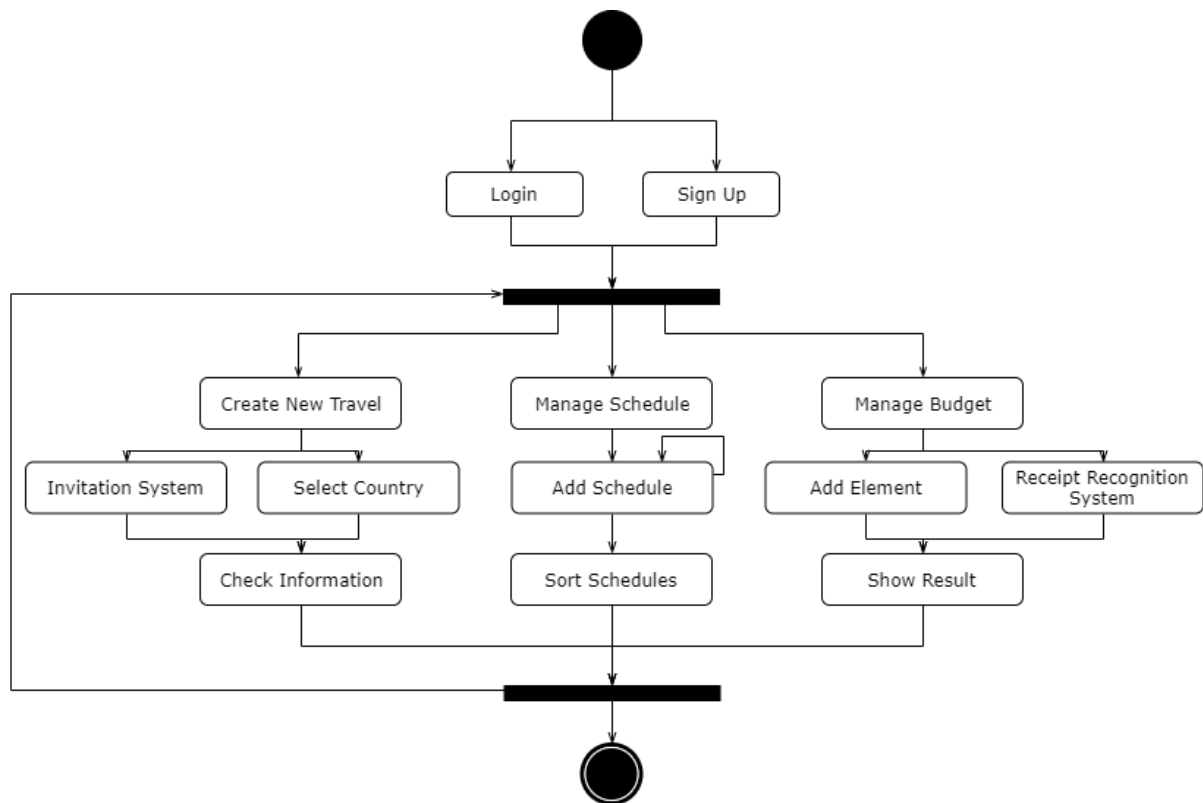


Diagram 8: Overall Process Diagram

7.2 Interaction Models

A. Use Case Diagram

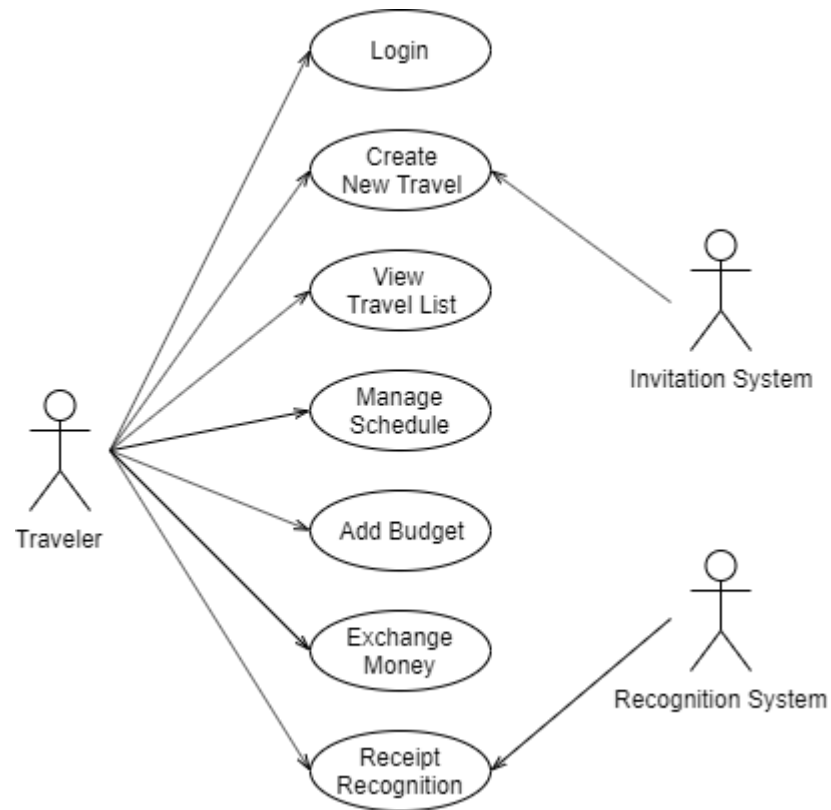


Diagram 9: Use Case Diagram

B. Tabular Description of Use Case Diagram

1. Login

Use Case	Login
Actor	User, Database(User)
Description	접속한 사용자가 시스템에 등록되어 있는 유저 정보와 일치하는지 판단하여 서버로부터 인증을 받는 과정이다.
Trigger	사용자가 ID와 PW를 입력한다.
Success Response	사용자 ID가 데이터베이스에 존재하고, PW의 Hash가 데이터베이스의 값과 일치하면 사용자 정보와 접속 권한을 포함한 객체를 돌려준다.
Failure Response	사용자 ID가 데이터베이스에 없을 경우 회원가입 페이지로 이동하며, PW가 일치하지 않을 경우 오류 메시지를 출력한다.

Table 14: Login Use Case

2. Create New Travel

Use Case	Create New Travel
Actor	User, Database(Travel)
Description	접속한 사용자가 새로운 여행 스케줄을 만드는 과정이다.
Trigger	여행시작일, 종료일, 여행국가, 여행 도시, 여행 일행을 검색 및 선택한다.
Success Response	여행전체목록에서 생성한 여행 탭이 생긴 것을 볼 수 있다.
Failure Response	필수입력란을 입력하지 않거나, 존재하지 않는 친구를 검색한 경우, 새로운 여행 객체는 생성되지 않으며, 오류 메시지를 출력한다. 또한, 네트워크 통신상태가 좋지 못하여, 새 여행이 등록이 되지 않을 수 있다. 이 경우, 네트워크 연결을 확인하고 다시 시도해주세요라는 경고 메시지가 출력된다.

Table 15: Create New Travel Use Case

Requirements Specification

3. View Travel List

Use Case	View Travel List
Actor	User, Database(Travel)
Description	접속한 사용자가 현재시점까지 생성한 모든 여행들을 리스트 형식으로 볼 수 있다.
Trigger	여행목록 탭 클릭
Success Response	모든 여행리스트가 사용자에게 슬라이드 형식으로 보여지며, 각각의 여행탭 안에는 사용자가 등록한 세부 여행 일정 이름과 여행지, 날짜, 동행자 등 필수 정보들이 보여진다.
Failure Response	만약 등록된 여행이 하나도 존재하지 않으면, 서버에 미리 준비한 여행 사진과 함께 감성적인 문구로 사용자에게 여행욕구를 자극한다. 또한, 네트워크 통신상태가 좋지 못하여, 여행 목록을 받아오지 못할 수 있다. 이 경우, 네트워크 연결을 확인하고 다시 시도해주세요라는 경고 메시지가 출력된다.

Table 16: View Travel List Use Case

4. Manage Schedule

Use Case	Manage Schedule
Actor	User, Database(Schedule)
Description	사용자가 선택한 특정 여행의 세부 일정을 보여주고, 세부 일정을 추가할 수 있다.
Trigger	전체 여행목록에서 특정 여행을 선택하여 눌러서 나온 화면에서 여행일정 제목, 장소를 기입한다.
Success Response	특정여행의 세부일정 한 개가 생성이 되며, 이는 전체여행리스트 화면에 보이는 여행슬라이드탭들 화면에도 작은 글씨로 표시된다.
Failure Response	네트워크 통신상태가 좋지 못하여, 세부 일정이 등록이 되지 않을 수 있다. 이 경우, 네트워크 연결을 확인하고 다시 시도해주세요라는 경고 메시지가 출력된다.

Table 17: Manage Schedule Use Case

Requirements Specification

5. Add Budget

Use Case	Add Budget
Actor	User, Database(Budget)
Description	사용자가 선택한 특정 여행의 예산, 지출내역을 추가한다.
Trigger	전체 여행목록에서 특정 여행을 선택하여 눌러서 나온 화면에서 예산관리 탭으로 이동한다. 생성된 일정들에 대하여 예산과 지출내역을 기입할 수 있다.
Success Response	예산과 지출내역을 정상적으로 모두 기입하였다면, 여행기간 전체 예산과 현재까지 지출비율, 날짜별 지출내역 등이 자동으로 계산되어 표시된다.
Failure Response	네트워크 통신상태가 좋지 못하여, 예산, 지출 등록이 되지 않을 수 있다. 이 경우, 네트워크 연결을 확인하고 다시 시도해주세요라는 경고 메시지가 출력된다.

Table 18: Add Budget Use Case

6. Exchange Money

Use Case	Exchange Money
Actor	User
Description	사용자가 기입한 예산, 지출내역 등을 실시간 환율을 적용하여 바꿔준다.
Trigger	유저는 Add Budget화면에 있는 실시간 환율 적용 버튼을 누른다.
Success Response	예산, 지출내역에 기입한 값들이 적용된 환율의 외화값으로 바뀐다.
Failure Response	네트워크 통신상태가 좋지 못하여, 서버에서 실시간 환율을 받아오지 못할 수 있다. 이 경우, 네트워크 연결을 확인하고 다시 시도해주세요라는 경고 메시지가 출력된다.

Table 19: Exchange Money Use Case

7. Receipt Recognition

Use Case	Receipt Recognition
Actor	User
Description	카메라/앨범을 통하여 영수증의 총계값을 인식

Requirements Specification

	하여 자동으로 인풋폼에 기입한다.
Trigger	예산 추가하기 화면에 있는 영수증 자동인식 버튼을 누른다. 카메라/앨범화면이 뜨고, 정상적으로 영수증 사진을 찍거나 선택하면 된다.
Success Response	예산 관리 추가하기에 자동으로 영수증의 총계값이 기입된다.
Failure Response	영수증 사진이 인식되지 않으면, 인식이 되지 않습니다. 재촬영을 하거나 직접 기입해주세요 라는 메시지가 출력된다.

Table 20: Receipt Recognition Use Case

7.3 Behavioral Models

이 장에서는 본 시스템에서 가장 중점으로 두는 하위 시스템인 '영수증 자동인식'과 '실시간 환율적용'의 Data flow diagram(DFD)를 제시한다.

A. 영수증 자동인식 DFD

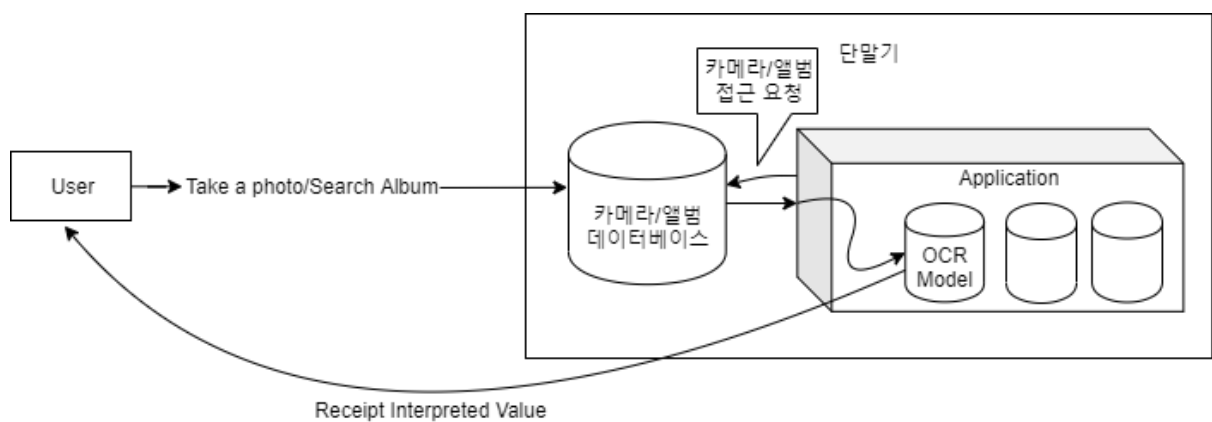


Diagram 10: Data Flow Diagram of Receipt Recognition

B. 실시간 환율적용 DFD

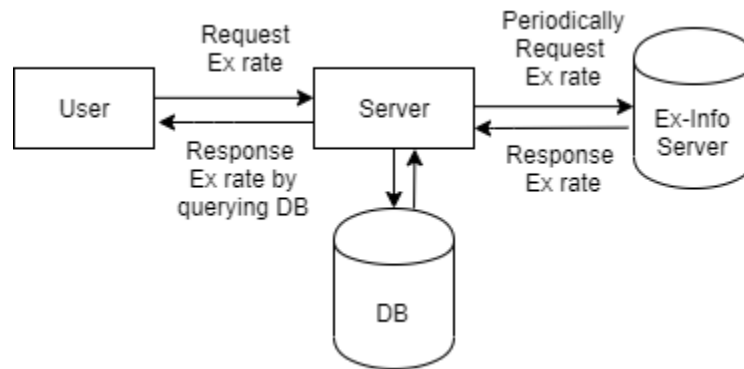


Diagram 11: Data Flow Diagram of Real-time Ex-rate

8. System Evolution

8.1 Objective

이 챕터에서는 시스템이 배포된 이후 운용되며 발생할 수 있는 시스템에 관련된 다양한 변화에 대해 예상하고, 그에 맞춰 가능한 대응에 대해 설명한다. 시스템 주변 환경부터 기술의 발전, 사용자의 요구사항에 대한 변화들이 일어날 수 있으며, 이로 인한 시스템 배포 이후에 필요한 변경 사항들을 미리 예측하고 적용시켜 시스템 변경 소요를 감소시킨다.

8.2 Evolutions

A. Travel Recommendation

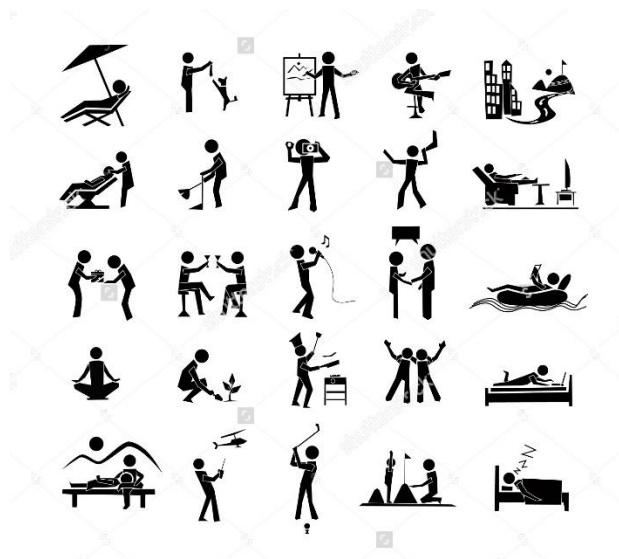


Figure 10: Travel Recommendation

Eco-travel은 사용자가 여행을 하고 있을 때에 필요한 기능만을 제공한다. 이를 해결하기 위해 진행중인 여행이 없는 사용자에게 관광 명소와 테마 등을 추천하는 기능을 추가한다. 사용자가 여행을 하고 있지 않더라도 시스템을 계속 사용할 수 있게 하며, 관광지나 항공사,

Requirements Specification

특정 음식점을 위해 광고를 하는 기능으로도 활용 가능하다. 여행을 진행중인 사용자의 경우 진행중인 일정의 위치를 이용해 그곳의 명소와 유명한 음식점, 액티비티 등을 안내해 사용자가 더 편리하게 여행할 수 있도록 한다.

B. Search Nearby Companions

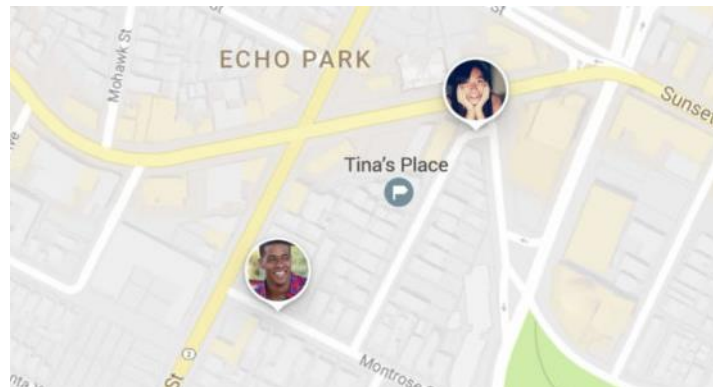


Figure 11: Search Nearby Companions

2인 이상의 동반 여행이 여전히 많지만 1인 여행객의 수 또한 갈수록 증가하고 있는 추세이다. 이러한 1인 여행객을 위해 여행 중에 위치 정보를 기반으로 근처의 1인 여행객을 찾아주는 기능을 제공한다. 동행을 찾고 있는 여행객에 한해 나타나며 매칭이 되었을 경우 같이 여행하는 기간 동안의 정보는 기존의 그룹 여행과 마찬가지로 공유할 수 있다. 이를 통해 사용자에게 폭 넓은 여행 경험을 제공할 수 있다.

C. Automatically Setting Budget

Requirements Specification



Figure 12: Schedule Prototype

등록된 여행 일정에 지출을 기입하지 않아도 필요한 예상 지출을 계산해주는 기능이다. 여행지의 환율과 물가, 다른 사용자가 기입한 일정과 지출 데이터를 활용해 계산한다. 사용자가 충분한 예산을 가지지 못했거나 예산에 맞춰서 일정을 정하기 어려울 경우 일정을 먼저 정한 후 예상되는 금액을 보고 일정이나 예산을 조절할 수 있다.

D. Oauth Implementation

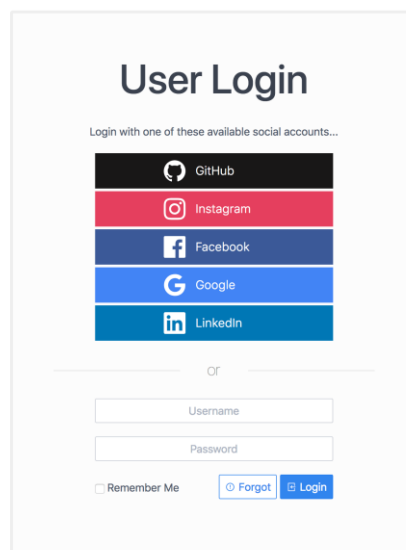


Figure 13: Oauth Implementation Prototype

페이스북, 인스타그램, 페이스북, 구글, 카카오, 네이버 등 한국인 사용자들이 많이 쓰는 SNS SIGNUP/LOGIN API를 적용함으로써 사용자 편의를 도모한다.

E. Kakao-talk Friends Invitation



Figure 14: Kakao-talk Friends Invitation

서비스가 활성화되면, 카카오톡과 제휴를 맺어서, 카카오편지장기능에서 보는 것과 같이 카카오편지장 친구초대기능 사용을 계약할 수 있다. 현재의 개발목표로 하고 있는 자체서버내에서의 친구보다는 한국인들이 가장 선호하는 SNS를 적용시켜 사용자 편의를 도모한다.

F. Add Comment/Re-comment on schedule

여행기록으로서의 역할을 충실히 하려면 여행 순간순간의 여행자가 느끼는 일정관리에서의 생각과 심정을 글로 남길 수 있고, 동행자들끼리 공유가 가능하므로 실시간 댓글을 다는 기능을 추가하여 사용자가 앱에서 머무를 수 있도록 자체 콘텐츠 생성을 유도한다.

Requirements Specification

9. Appendices

이번 챕터에서는 앞서 본문에서 다루지 못한 시스템 개발에 관련된 세부 정보를 다룬다. 시스템 운영 환경에 필요한 하드웨어 요구사항, 시스템에 사용되는 데이터베이스의 개략적인 구조 등에 대해 기술한다.

9.1 Hardware Requirements

본 시스템은 React-native의 기술을 이용하여 여행객들의 UI/UX 사용성을 높인 모바일 어플리케이션을 목표로 한다. 안드로이드, ios 등 운영체제가 다른 단말기에서 이 시스템을 구동시켰을 때, UI/UX면에 있어서 큰 차이가 있어서는 안된다. 사용성을 극대화시키기 위하여 가능하다면 UI에 있어서 애니메이션적 요소를 사용할 것이므로, 사용자는 ES6를 지원하는 브라우저를 사용하는 것을 권장한다. 또한, 사용자는 같이 여행을 가는 친구와 함께 소켓통신을 하므로, 같은 여행그룹 사용자들의 단말기는 인터넷에 연결되어야 실시간 동기화를 보장받을 수 있다.

9.2 Database Requirements

본 시스템은 Mysql을 활용한 관계형 데이터베이스를 사용하여 데이터를 저장 및 관리한다. 데이터베이스 내의 엔티티들은 아래와 같다. 세부적인 데이터베이스 설계는 Design Architecture 문서에서 다루게 된다.

Requirements Specification

A. User

아이디, 비밀번호, 이름, 닉네임 등의 사용자 정보를 담고 있는 엔티티

B. Travel

사용자가 추가한 여행에 대한 정보를 담고 있는 엔티티

C. Schedule

사용자가 추가한 여행의 세부일정에 대한 정보를 담고 있는 엔티티

D. Budget

사용자가 추가한 여행의 예산, 지출내역에 대한 정보를 담고 있는 엔티티

E. Travel-schedule mapping table

M:N관계를 맺고 있는 Travel모델과 Schedule모델을 이어주는 매핑 테이블 엔티티

F. Nations/cities

전세계의 국가와 도시들의 정보를 담고 있는 엔티티

10. Index

10.1 Tables

Table 1: Glossary	14
Table 2: Login Function.....	28
Table 3: Signup Function.....	29
Table 4: Add New Travel Function	30
Table 5: Search Destination Function	30
Table 6: Search Friends Function	31
Table 7: Show Travel List Function.....	31
Table 8: Show Travel Detail Function.....	32
Table 9: Show Travel Schedule Function	32
Table 10: Add(Edit) Travel Schedule Function	33
Table 11: Show Budget Management Function	34
Table 12: Show Pay-map Function.....	34
Table 13: Receipt Recognition Function.....	35
Table 14: Login Use Case	44
Table 15: Create New Travel Use Case.....	44
Table 16: View Travel List Use Case.....	45
Table 17: Manage Schedule Use Case	45
Table 18: Add Budget Use Case	46
Table 19: Exchange Money Use Case.....	46
Table 20: Receipt Recognition Use Case.....	47

10.2 Figures

Figure 1: 국민여행 실태조사.....	8
Figure 2: 해외여행객 수 추이변화	9
Figure 3: 시스템 목업 - 일정 실시간 공유	10
Figure 4: 시스템 목업 - 영수증 카메라 자동인식	11
Figure 5: Travel Schedule Prototype	16
Figure 6: Budget Management Prototype	17
Figure 7: 영수증 카메라 자동인식(2).....	18
Figure 8: Schedule Map Prototype.....	18
Figure 9: Add Travel Prototype.....	19
Figure 10: Travel Recommendation	49
Figure 11: Search Nearby Companions	50
Figure 12: Schedule Prototype	51
Figure 13: Oauth Implementation Prototype.....	51
Figure 14: Kakao-talk Friends Invitation.....	52

10.3 Diagrams

Diagram 1: Signup&Login Process.....	15
Diagram 2: Frontend Architecture	23
Diagram 3: Socket Communication.....	24
Diagram 4: Backend Architecture.....	25
Diagram 5: Flowchart of Getting Receipt Info	26
Diagram 6: Exchange Rate System	27
Diagram 7: Overall Context Diagram	41
Diagram 8: Overall Process Diagram.....	42
Diagram 9: Use Case Diagram	43
Diagram 10: Data Flow Diagram of Receipt Recognition	47
Diagram 11: Data Flow Diagram of Real-time Ex-rate.....	48

11. References

한국관광통계(한국관광공사 발표) :

http://www.index.go.kr/potal/main/EachDtlPageDetail.do?idx_cd=1655

프로토타입참고 : 세이브트립(어플리케이션)