Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 18/03/2014 | 1.1 | Added document overview and system overview. | Elvin Ege Şenoymak |
| 18/03/2014 | 1.1 | Added  abbreviations, conventions and system architecture. | Begüm Tokuyucu |
| 18/03/2014 | 1.1 | Added User Requirements Specification | Pamir Çevikoğulları |

TABLE OF CONTENTS

Revision History 1

INTRODUCTION 3

Document overview 3

System overview 3

Abbreviations 3

References 3

Conventions 3

REQUIREMENTS 5

User Requirements Specification 5

System Architecture 5

Use Cases and Usage Scenarios 6

System Requirements Specification 6

# INTRODUCTION

## Document overview

This document presents the software requirements specifications of ÇetinApp software development project.

It includes;

* User requirements and System requirements,
* System Architecture,
* Use cases and usage scenarios.

## System overview

Our app needs a MySQL database and a server to work. It will work on any web browser, Android 3.0+ and iOS 7.0+. In order to communicate, all platforms will have two main job. To send data, and to receive data from the database. Devices will continuously listen for incoming messages, and synchronously will send messages which saves them to the database. So this way, each device will work together in harmony.

ÇetinApp’s supported language will be English. Users should be able to understand simple English in order to use our app with its full functionality. Also users are requested to signed up and authenticated to use the app.

## Abbreviations

SDP: Software Development Plan

SRS: Software Requirements Specification

MySQL: My Structed Query Language

REQ: Requirements

SYSREQ: System Requirements

CONV: Convention

## References

## Conventions

Requirements listed in this document are constructed according to the following structure:

Font – Times New Roman, size 12

Main headings, Bold size 16

Sub headings, Bold size 14

Requirements structure in this document should be like;

Requirement Id

Requirement title

Requirement description

Requirement version

System architecture structure in this document should be like;

Name of the Project/Project Type/Document Version

Function

Description

Inputs

Source

Output

Destination

# REQUIREMENTS

## User Requirements Specification

**SRS-REQ-001**

Sending messages

Users shall send message to a target conversation.

V1.0

**SRS-REQ-002**

Receiving messages

Users shall receive messages from conversations that they are participating in.

V1.0

**SRS-REQ-003**

Chatlog

Chatlog shall be saved for users.

V1.0

**SRS-REQ-004**

Chatlog request

Chatlog of conversations shall be sent to the user email upon request.

V1.0

**SRS-REQ-005**

Group Chat

Users shall chat as a group.

V1.0

## System Architecture

ÇetinApp/SRS/1.0

Function Conversation backup

Description Retrieves all messages of the conversation and sends the user via email.

Inputs User ID, Conversation ID

Source Client

Output Conversation that will be sent by mail

Destination User’s mail address

ÇetinApp/SRS/1.0

Function Sending message to database

Description Message will be requested from the client and saved to the database.

Inputs User ID, Conversation ID, Message

Source Client

Output Success/Fail message

Destination Database

ÇetinApp/SRS/1.0

Function Requesting message from database

Description Requests for new messages will be made to database

Inputs Conversation ID

Source Database

Output Source User ID, Message

Destination Client

## Use Cases and Usage Scenarios

Include here the relevant UML use case diagrams and sequence diagrams that depict the usage scenarios of the system.

## System Requirements Specification

This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. They should be separated in a subsection. Interfaces to other systems may be defined. Activity diagrams and sequence diagrams should depict the protocols utilized in such interfaces.

SRS-XXX-030.2 SAMPLE

Patient data

XXX ensures that the displayed patient data are the same as read in the input files.

The patient’s data are:

* Name,
* Date of birth,

V1.0

SRS-XXX-040 SAMPLE

Application logs

XXX generates a log file containing:

* The state of the application and the steps performed to reach that state,
* The possible error logs, if any.

V1.0