System Design Document for TheraComm

Prepared by

Christian John	200360001
Iden Ellia	200370502
Joe Samano	200365260

Version History

Document History

Version #	Implemented by	Revision Date	Approved by	Approval Date	Reason
1	Christian John Joe Samano Iden Ellia	15/03/2019		25/03/2019	Draft

Table of Contents

1 Introduction

1.1 Project Overview

TheraComm is an iOS-based mobile application that provides patients a convenient way to communicate with their therapist. TheraComm incorporates the core features of an email system and a chat system. Thus, allowing the patient to not only communicate with the therapist, but also have the communication in real time.

1.2 Purpose

The purpose of this document is to provide a description of the design of TheraComm to allow for software development to proceed with an understanding of what is to be built and how it is expected to built. Furthermore, this document is intended for developers to understand how TheraComm is built.

1.3 Design Goals

The design goals for this project are the following:

- The system must be an iOS-based application built in Swift programming language.
- The implementation of the email system must follow how it is implemented using the API used in the Online Therapy Unit's email system.
- The implementation of the chat system must use a third-party API particularly Chatkit.
- The user interface of the application must be user-friendly in terms of ease of use and visually appealing.
- The system must be extensible.

1.4 Definitions, Acronyms, and Abbreviations

API An application program interface (API) is a set of routines, protocols, and tools for building software applications.

IDE An integrated development environment (API) is a software application for programmers to use for software developments.

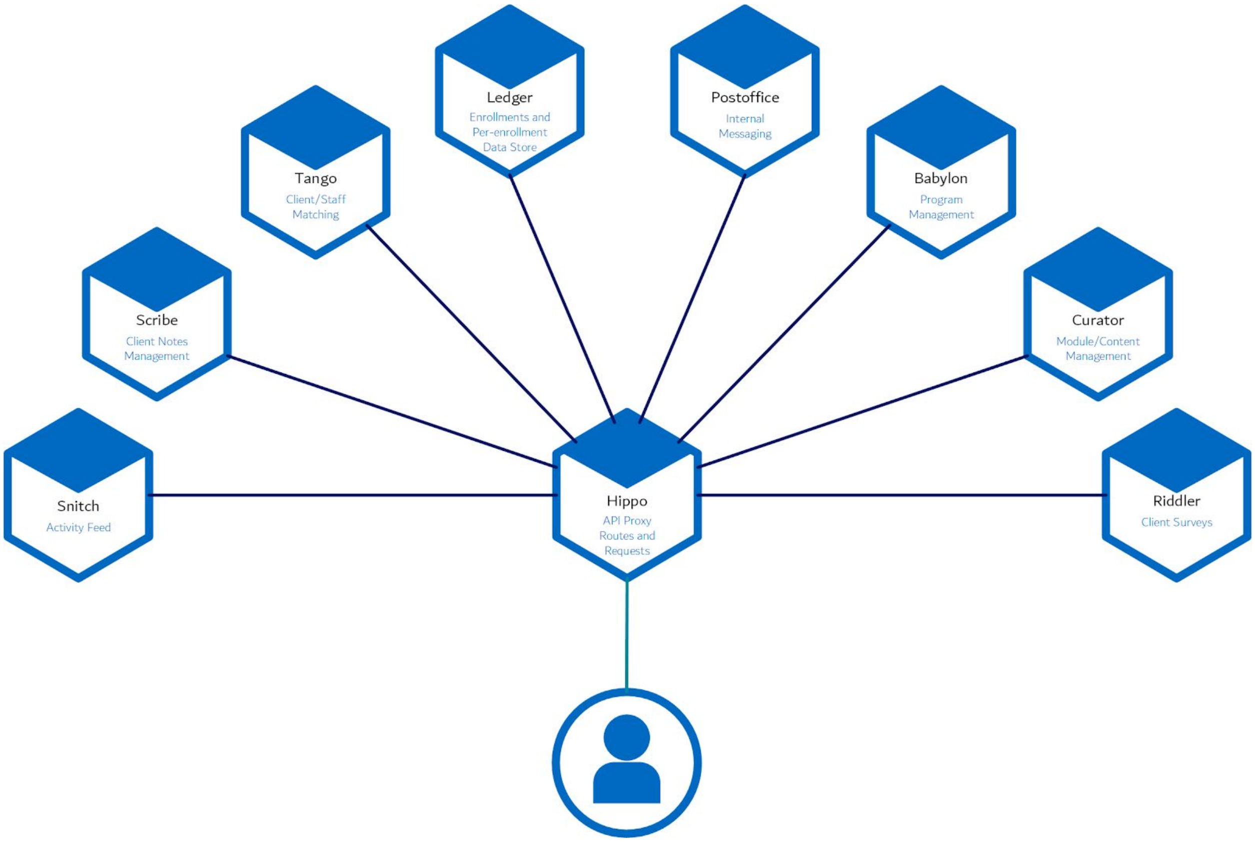
1.5 References

System Design Document Template.(2003, September 30).Retrieved from http://ee.hawaii.edu/~tep/EE467/BDLecs/html/sdd.htm

2. Current Software Architecture

The system being used by the Online Therapy Unit is implemented in a web application. Their service platform is based on microservice architecture. The structure of the architecture is a collection of loosely coupled services that each provides an API. These services are the following:

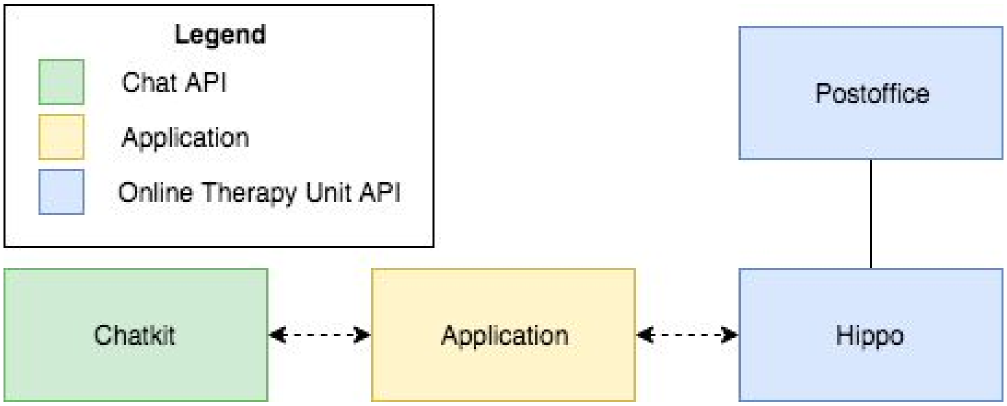
- Hippo - API Proxy
- Snitch - Activity Feeds
- Scribe - Client notes management
- Tango - Client/staff matchings
- Ledger - Program enrollments and per-enrollment client data store
- Postoffice - Internal messaging system
- Babylon - Program management
- Curator - Module/content management
- Ridler - Client surveys
- Trumpet - Site-wide system notifications aka broadcasts



3. Proposed Software Architecture

3.1 Overview

This system is separated into three components: Application, Chat API, and the Online Therapy Unit API. The application is the user interface that will be containing the email and chat services in order to interact with the users. The Chat API is provided by a third-party application called Chatkit to provide us with the core features of the chat part of the application. The Online Therapy Unit API provides the email features of the email part of the application.



3.2 Subsystem Decomposition

3.2.1 Application

The application is built using the Swift language and using an IDE called Xcode. Furthermore, the application is going to be an iOS-based application. The main responsibility of this component is to provide the user with an interface in order to use the email and chat features of the application.

3.2.2 Online Therapy Unit API

The Online Therapy Unit API's main responsibility is to provide the application with the email features. In order to gain access and interact to Postoffice, an API for internal messaging and email system, the application needs to communicate with Hippo, a public-facing API Proxy/Gateway.

3.2.3 Chat API

The Chat API is provided by Chatkit, a third party extensible API to build in-app messaging. The main responsibility of this component is to provide the application with the chat features. These features include being able to create rooms to have one-on-one or group chats, create or send a chat message, push notifications, typing indicator, and read cursors.

3.2.3.1 Advantages of using Chatkit

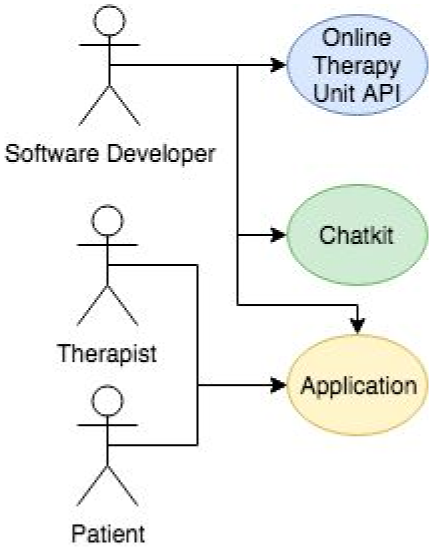
- Good Documentation
- The Sandbox Plan is free which allows up to 1000 users per month, 500,000 messages per month, and contains the core features necessary for a chat system.
- Can be built with multiple languages such as Swift, Java, Javascript, etc.
- No need to make any changes in the Online Therapy Unit API to make Chatkit work.

3.2.3.2 Disadvantages of using Chatkit

- Costly when upgrading plan
- End-to-end encryption is still in development

3.3 Access Control

There are going to be three people who are involved in the system: software developer, therapist, and the patient. The software developer will be managing the system, while the therapist and patient will only be able to access the application. The software developer can edit the Online Therapy Unit API in order to make necessary changes for the email system. Furthermore, the software developer can manage the rooms and users using Chatkit for the chat system. The software developer can also make changes in the user interface of the application. The therapist and patient can only interact and make use of the services in the application.


Software Developer
Therapist
Patient
Online Therapy Unit API
Chatkit
Application