Text To Pay

Project Plan

Version 1.4

**Team Name: Four Musks**

**Team Members**:

| **Name** | **SJSU ID** |
| --- | --- |
| Utsav Rawat | 016664466 |
| Jaya Krishna Thupili | 016722277 |
| Nihal Kaul | 016697512 |
| Ninad Marathe | 016597503 |

**Repository URL:**

[**https://github.com/sjsu-4musks/text-to-pay-app**](https://github.com/sjsu-4musks/text-to-pay-app)

**https://github.com/sjsu-4musks/text-to-pay-api**

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 02/04/2023 | 1.1 | Requirement Analysis. Exploring tech stack, project architecture and components best suited for the application. | Utsav, Nihal, Jaya, Ninad |
| 03/15/2023 | 1.2 | Created basic templates to be used for all the web pages. | Utsav, Nihal, Jaya, Ninad |
| 04/08/2023 | 1.3 | Created API backend for all the layouts and developed the front end. | Utsav, Nihal, Jaya, Ninad |
| 05/05/2023 | 1.4 | Integration and deployment of the code in AWS cloud services. | Utsav, Nihal, Jaya, Ninad |

Table of Contents

1. Introduction 4

1.1 Purpose of this document 4

1.2 Intended Audience 4

1.3 Scope 4

1.4 Definitions and acronyms 4

1.4.1 Definitions 4

1.4.2 Acronyms and abbreviations 5

1.5 References 5

2. Background and Objectives 5

3. Organization 5

*3.1* Project group 5

3.2 Customer 6

4. Development process 6

*5.* Deliverables 6

6. Project risks 6

7. Communication 7

7.1 Canvas 7

7.2 Git 7

8. Project plan 7

8.1 Time schedule 7

8.1.1 Remarks 7

8.2 Test plan 9

8.2.1 Testing Remarks 10

9. References 10

# Introduction

The concept of 'Text to Pay' or SMS payment, as it's commonly known, is a straightforward, innovative solution. It empowers consumers to make payments via texting on their smartphones. The process is simple: a business sends a text to a client's phone, stating they can make a payment, and the client has the option to text back "yes" to complete the transaction.

## Purpose of this document

The purpose of the document is to serve as a complete guide detailing the "Text to Pay" project which allows you to get paid from any phone number, extending the scope of your customer base. The document includes objectives, organization, development process, and deliverables. It will also address potential risks and include relevant references.

## Intended Audience

This document shall be used in all phases of the project as a guideline. The primary audience of this document includes:

* project supervisor
* project leader
* team members

## Scope

The scope of the document covers the development and implementation of the "Text to Pay" system. It will not delve into similar payment systems or the marketing strategies for promoting this system. The overview includes the objectives of the project, organization of the project team, development process that is going to be used during the project, assessment of possible risks, communication used between project stakeholders, and project plan that includes time schedule and activity plan.

## Definitions and acronyms

### Definitions

| **Keyword** | **Definitions** |
| --- | --- |
| Text to Pay | The name of the project |
| Project Leader | A person in charge of organizing the team and communicating with the project supervisor |
| Team Member | An active member of the team responsible for making the job done |
| Milestone | A time in a project that marks the end of a project phase or the completion of an important deliverable. |
| Git | The version control system that will be used in this project |
| Scrum | An iterative and incremental agile software development method for managing software projects and product or application development |
|  |  |
| Twilio | Twilio Messaging Services API is a Programmable Messaging used to send high volumes of messages |
| Stripe | Stripe is a suite of APIs powering online payment processing and commerce solutions for internet businesses |
| Scrum Sprint | The basic unit of development in Scrum |

## References

<https://stripe.com/docs/api>

<https://www.twilio.com/docs/sms/quickstart/node>

# Background and Objectives:

In this section, the problem businesses face is outlined, and the solution the project aims to provide is explained. The objective is to develop a system that leverages the speed and convenience of text messaging to create an efficient, secure, and user-friendly payment method.

The main objectives of the 'Text to Pay' application are as follows:

* Streamline Payments.
* Improve Customer Engagement
* Increase Conversion Rates
* Enhance Customer Retention
* Facilitate Onboarding of New Customers
* Boost Security

# 

# Architecture & High-Level Design

A screenshot of a computer

Description automatically generated with low confidence

**Use Case Sequence Diagram**

**A picture containing text, screenshot, font, number

Description automatically generated**

# Organization

San Jose state University- Master of Science In Software Engineering – Cloud-14 Cohort

## Project group

Elite

| **Name** | **Initials** | **Responsibility (roles)** |
| --- | --- | --- |
| Nihal | Kaul | Frontend-Backend code development and AWS deployment. |
| Utsav | Rawat | Frontend-Backend code development and AWS deployment. |
| Jaya | Thupili | Frontend-Backend code development and AWS deployment. |
| Ninad | Marathe | Frontend-Backend code development and AWS deployment. |

## Customer

The 'Text to Pay' application can be beneficial to a wide range of customers. Here's a broad categorization of our target customers:

* Retail Businesses
* Restaurants and Food Delivery Services.
* Subscription Services.
* Service Providers.
* Events and Entertainment.
* Non-profit Organizations.

# Development process

'Text to Pay' is a serverless application running on AWS Lambda. The sign-up and sign-in form is authorized and authenticated using Auth0 login via a Gmail account. The backend API are deployed in the lambda function and the API gateway is used to trigger the APIs. The backend code is written in Nodejs. The front end is designed using Reactjs. For front-end deployment, we leveraged Amazon S3 and integrated it with Cloudfront. S3 ensures the data is highly scalable, highly available, fast, and secure. CloudFront caches the content across multiple distribution servers located close to our users, significantly reducing latency and providing a smoother user experience.

# Deliverables

1. PROVIDER

* Seamless Integration: The 'Text to Pay' system will seamlessly integrate with the business's existing systems, providing a smooth transition and minimal disruption to operations.
* Training Materials: Providers will receive comprehensive training materials and support to understand how to use and manage the 'Text to Pay' system effectively.
* Marketing Assistance: Businesses will be provided with tools and strategies to market the new payment option to their customers, including templates for text messages, promotional materials, and more.
* Data Analytics: Providers will have access to valuable analytics regarding customer purchasing behavior, response times, and other metrics that can help refine their strategies and boost sales.
* 24/7 Customer Support: A round-the-clock customer service team will be available to assist providers with any issues or queries they might have regarding the 'Text to Pay' system.
* Security Assurance: The system will deliver secure transaction processing, with robust data encryption and fraud prevention measures in place.

2. CONSUMER

* User-friendly Interface: The 'Text to Pay' system will offer a simple and intuitive user interface, making it easy for customers to understand and use.
* Secure Payments: Customers will be able to make secure payments without worrying about their personal financial information being compromised.
* Fast Transactions: The system promises quick transaction times, making it a convenient option for customers.
* Clear Instructions: Customers will receive clear instructions on how to use the 'Text to Pay' system, including how to register, how to make payments, and how to securely store their payment information.
* Customer Support: Customers will also have access to a customer service team to resolve any issues or answer questions about the 'Text to Pay' system.
* Flexible Payment Options: Customers will have the flexibility to add multiple payment methods and choose their preferred option for transactions. Cancel scheduled services

# Project risks

| **Possibility** | **Risk** | **Preventive action** |
| --- | --- | --- |
| Security Risks | SMS Spoofing | Following MFA approach, by integrating Auth0 and CAPTHA along with SMS rate limits during user registration. |
| Scalability issue | Service outage and high expenses | We have deployed our backend APIs on AWS Lambda making it a serverless architecture. Lambda is inexpensive in comparison with EC2 and other instances. It has an SLA of at least 99.95% . |
| Disaster Recovery |  | Frontend is served via AWS CloudFront and backend is served via AWS Lambda.  CloudFront is configured with origin groups served by AWS S3 buckets from 2 different regions. This configuration ensured faster responses via cloudfront and DR resilient of frontend when one of the serving regions is down. |

# Communication

We held bi-weekly meetings, with each sprint starting on a Monday and lasting two weeks. Each team member committed to spending 20 hours per sprint, or 10 hours per week, on project activities. We also decided to rotate the role of Scrum Master for each sprint, offering each team member a chance to lead and bring unique ideas to the table.

## Collaboration

We discussed our progress and dealt with dependence concerns and other roadblocks during the call. To keep track of our user stories and manage the project, we used Project Dashboard.

## Git

All source code and finished documentation will be uploaded to GitHub repository.

**Repository URL:**

[**https://github.com/sjsu-4musks/text-to-pay-app**](https://github.com/sjsu-4musks/text-to-pay-app)

**https://github.com/sjsu-4musks/text-to-pay-api**

# Project plan

## Time schedule

| **Id** | **Milestone**  **Description** | **Responsible Dept./Initials** | **Finished week** |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Plan** | **Forecast** |  | **Actual** |
|  |  |  |  | **Week** | **+/-** |  |
| 1 | Sprint 1  Project planning and setup, development environment setup, research, and requirements gathering | Nihal, Jaya | Week2 | Week2 |  | Week2 |
| 2 | Sprint 2  Front-end development, design the user interface, and authentication setup using Auth0. | Nihal, Utsav, Jaya, Ninad | Week2 | Week2 |  | Week2 |
| 3 | Sprint 3  Integration of Stripe Payment Gateway, development of API, and serverless backend setup on AWS Lambda. 2 weeks | Nihal, Utsav, Jaya, Ninad | Week2 | Week2 |  | Week2 |
| 4 | Sprint 4  Backend integration with the front end, testing and bug fixes. | Nihal, Utsav, Jaya, Ninad | Week2 | Week2 |  | Week2 |
| 5 | Sprint 5  Deployment and configuration of the AWS services, implementation of Twilio API for SMS integration, and user acceptance testing. | Nihal, Utsav, Jaya, Ninad | Week2 | Week2 |  | Week2 |
| 6 | AWS configuration for S3, route 53, CloudFront. | Nihal, Utsav, Jaya, Ninad | Week3 | Week3 |  | Week3 |
| 7 | Integration and deployment of the code in AWS for AWS Elastic Beanstalk. | Nihal, Utsav, Jaya, Ninad | Week4 | Week4 |  | Week4 |

## 

## Test plan

| **Test No.** | 001 | **Phase:** | 1 | **Author:** | <username> | Date: |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Category:** | | Functional tests | | | [Jaya Krishna Thupili](mailto:jayakrishna.thupili@sjsu.edu) | 10 May 2023 |
| **Software Product:** | | Text to Pay | | | |  |
| **Test Title:** | | Backend API functional tests | | | | |
| **Test Purpose:** | | Test backend APIs | | | | |
| **Test Setup:** | | Wrote Postman test suites | | | | |
| **Prerequisites:** | | Postman software | | | | |
| **Procedure:** | | Use a postman docker image and run the postman container by passing the collections of test suits and the corresponding environment variables required for the same. | | | | |
| **Checks:** | | Wrote a series of functional tests in postman, these tests made a series of backend api calls and then tested for the schema, the json response, the status code of each api calls, the expected string validation, checking for valid error messages, checking for the list of mandatory attributes in the response. | | | | |
| **Expected Results:** | | All the functional tests should pass | | | | |
| **Result:** | | We got all the Functional tests passed and the result was available in the report | | | | |
| **Reason for Failure:** | | No failures | | | | |
| **Remarks:** | | Code ready to merge | | | | |

# References

Image reference used in Text to Pay are:

* Twillio logo (<https://www.stickpng.com/img/icons-logos-emojis/tech-companies/twilio-full-logo>)
* Stripe Logo (<https://www.stickpng.com/img/icons-logos-emojis/tech-companies/stripe-payment-logo>)
* MongoDB Atlas ([https://www.mongodb.com](https://www.mongodb.com/))
* AWS Resources logo([https://aws.amazon.com/architecture/icons](https://aws.amazon.com/architecture/icons/))

Tools used for diagrams:

* Draw.io
* creately