

Scrum Meeting Report to Track Progress

Note: A different group member should volunteer each scrum report to prepare and turn in this report. That way a particular member does not get overwhelmed doing all reports through the semester.

Group Name: WebCraft

Reporter: Elliot Fayman

Date: 03/02/2023

Starting Time: 5:30 pm

Ending Time: 7:00 pm

Participants: Elliot Fayman, Ryan Kao, Christina Mourad

1. Each member must individually answer the below three questions:

- a. What have you accomplished during the past two weeks?
 - a. **Elliot** – In the past two weeks, I helped create a new endpoint for our flask api which will allow us to ping the server in order to test for successful connection.
 - b. **Ryan** – Created API documentation with Flask that contains information about our API
 - c. **Christina** – Created a home screen on Flutter with an interactive button that opens a separate page
- b. What are you going to get done during the next two weeks?
 - a. **Elliot** – This week I hope to integrate the twillio api into the flask application. I hope to be able to send sms messages from the flask api to a specified phone number. (TEX-17)
 - b. **Ryan** – Continue to add to API documentation as we further develop API and start working on the Twillio sms api integration. (TEX-18)
 - c. **Christina** – Edit the styling of the home screen and adding a second page. Begin working on Jira task (TEX-31)
- c. What obstacles do you need to be removed?
 - a. **Elliot** – The obstacle I might need to remove this week is the fact that the flask api has no production plans. We might need to decide how we are going to host the api so that integrating the features needed to host the api will be more seamless.
 - b. **Ryan** – Becoming more familiar with Twilio and preparing to integrate it into the api.
 - c. **Christina** – Learning more about SMS messaging and how to embed it inside the Flutter app

2. Provide a Burn-down list for current Sprint, i.e., list of user stories that are yet to be completed for the current sprint and user stories that have been completed for the current sprint.

(embed web link if your user stories are managed via an online platform, give me permission to view)

[TextPy - Agile board - Jira \(atlassian.net\)](#)

Burn-Down List for Current Sprint:

- **TEX-10** User Should be greeted with home page when app is first opened
- **TEX-17** User should be able to receive SMS messages from Flask application
- **TEX-21** User should be able to send SMS messages to the Flask application
- **TEX-25** User should be able to type and send messages in a different app view
- **TEX-18** Configure Twillio Account to be able to send SMS messages (In progress)
- **TEX-19** Add method to Flask api that sends sms message to a specific phone number (In progress)
- **TEX-31** Improve Home Screen Styling (In progress)
- **TEX-22** Configure Twillio to post to a specific Flask endpoint
- **TEX-23** Add Flask endpoint that processes Twilio Post

- **TEX-24** Use send SMS method to then send an SMS message upon receiving a user sent SMS
- **TEX-29** Create new view for Messaging
- **TEX-30** Create new Flutter UI that allows user to type and send text

3. Additionally, you are asked to provide a snapshot of Agile Task Board for your project:

Agile Task Board - Task board is used to track progress in agile for one particular or a set of user stories. Each user story would require a set of implementation tasks. Your task board should have the below set of information:

User Story: user story under consideration for current sprint.

To Do: Tasks that needs to be picked up.

In Progress: Tasks that are currently in progress, with an estimated date of when it would get completed.

To Verify: Tasks that needs to be verified/tested.

Done: Tasks that are completed.

You may maintain your 2. and 3. online. In that case you can embed the links here and give me permission to view your 2. and 3.

[TEX board - Agile board - Jira \(atlassian.net\)](#)

Projects / TextPy

TEX Sprint 2

Integrate Twilio API into flask api and create a new view for messaging in the twillio application

Search: [] Filters: [Epic] [Type]

USER STORY 4 ISSUES	TO DO 5 ISSUES	IN PROGRESS 3 ISSUES	VERIFY	DONE
<p>User Should be greeted with home page when app is first opened</p> <p>FLUTTER HOME SCREEN</p> <p>TEX-10</p>	<p>Configure Twilio to post to a specific Flask endpoint</p> <p>FLASK TWILLIO INTERFACE</p> <p>TEX-22</p>	<p>Configure Twilio Account to be able to send SMS messages</p> <p>FLASK TWILLIO INTERFACE</p> <p>TEX-18</p>		
<p>User should be able to receive SMS messages from Flask application</p> <p>FLASK TWILLIO INTERFACE</p> <p>TEX-17</p>	<p>Add Flask endpoint that processes Twilio Post</p> <p>FLASK TWILLIO INTERFACE</p> <p>TEX-23</p>	<p>Add method to Flask api that sends sms message to a specific phone number</p> <p>FLASK TWILLIO INTERFACE</p> <p>TEX-19</p>		
<p>User should be able to send SMS messages to the Flask application</p> <p>FLASK TWILLIO INTERFACE</p> <p>TEX-21</p>	<p>Use send SMS method to then send an SMS message upon receiving a user sent SMS</p> <p>FLASK TWILLIO INTERFACE</p> <p>TEX-24</p>	<p>Improve Home Screen Styling</p> <p>FLUTTER HOME SCREEN</p> <p>TEX-31</p>		
<p>User should be able to type and send messages in a different app view</p> <p>FLUTTER MESSAGING VIEW</p> <p>TEX-25</p>	<p>Create new view for Messaging</p> <p>FLUTTER MESSAGING VIEW</p> <p>TEX-29</p>			
	<p>Create new Flutter UI that allows user to type and send text</p> <p>FLUTTER MESSAGING VIEW</p> <p>TEX-30</p>			

4. Any other comments:

No additional comments.