

## MILESTONE 4



Team Lead :	<b>Cassie Sherman</b>	cassiesherman630@gmail.com
Backend Lead :	Ryan Hu	213edu@gmail.com
Frontend Lead :	Mariam Shahab	mariamshahab2@gmail.com
Database Lead :	Dylan Luong	dluong@mail.sfsu.edu
Github Master :	Sawara Bhattarai	sbhattarai@mail.sfsu.edu
Documentation Lead :	Arjun Bista	arjunbista428@gmail.com

## History Table

Milestones/Version	Date
M4 V1	05/10/2020
M3 V2	05/09/2020
M3 V1	04/23/2020
M2 V2	04/09/2020
M2 V1	03/26/2020
M1 V2	03/09/2020
M1 V1	03/05/2020

# Product summary

## LingoMingo - A Learning Community

1. Registered users shall be able to log in to account.
  - 1.1 Registered users shall enter an email or an username.
  - 1.2 Registered users shall enter a password.
  - 1.3 Registered users shall be able to reset password, if forgotten.
2. Registered users shall be able to log out to account.
  - 2.1 Registered users have to be logged in before being able to log out.
3. Registered users shall be able to delete the account.
  - 3.1 Register user's profile shall also be deleted.
  - 3.2 Register users shall be able to temporarily deactivate accounts.
  - 3.3 Register users shall be able to reactivate accounts.
4. Registered users shall be able to make changes in setting.
  - 4.1 Registered users shall be able to delete their account from the settings.
  - 4.2 Registered users shall be able to change their password.
5. Registered users shall be able to choose a language to learn.
  - 5.1 Registered users shall be able to change the language they want to learn.
  - 5.2 Registered users shall choose from a list of available languages on LingoMingo.
6. Registered users shall be able to choose to learn multiple languages.
  - 6.1 Registered users shall be able to learn more than one language.
  - 6.2 Registered users shall be able to teach language to others.
7. Registered users shall be able to choose icons for display
  - 7.1 Registered users' icons shall appear on their profile.
  - 7.2 Registered users shall be able change to a different icon.
8. Registered users shall be able to write a bio
  - 8.1 Bio shall be displayed in user's profile
  - 8.2 Registered users shall be able to update their bio

- 9. Registered users shall be able to add another user as a friend.
  - 9.1 Registered users shall be able to unfriend.
  - 9.3 Registered users shall be able to create a chat with a friend user.
  - 9.4 Registered users shall be able to look at friends' profiles.
- 10. Registered users shall be able to ask for the help.
  - 10.1 Registered users shall be able to ask help with the support team.
  - 10.2 Registered users shall be able to email support teams.
- 11. Registered users shall be able to submit data.
  - 11.1 Registered users shall be able to submit images in the chat.
  - 11.2 Registered users shall be able to submit text in the chat.
- 12. Registered users shall be able to make new posts.
  - 12.1 Registered users shall be able to edit posts.
  - 12.2 Registered users shall be able to delete posts.
  - 12.3 Registered users shall be able to post unlimited.

We plan to market and sell our service by providing a solution to the problem people didn't know they had while learning a language. Learning language through a course is repetitive, boring, and costly, not to mention hard to retain, but speaking to individuals is a whole new ballpark. Socially learning is one of the best ways to improve communication and that is exactly what our users need. We will promote this service using that ideology. Our product offers the unique service of learning a language through the words of an actual nativetongued individual. Not only do you have the awesome opportunity of meeting people from all around the world, but you'll also be learning while you are at it. All our user has to do is put the language they'd like to learn and they're set to go. Our user's matching system is effective and will match the best user for the individual according to their personal preference. There aren't many products offering the same service and for the user's convenience our product can be easily accessed not only in the mobile format but also in the web browser.

URL: [Lingomingo.app](https://lingomingo.app)

# Usability Test Plan

## Functions:

Setting Languages, Matching, Chat, Edit Profile, Posting

## Test Objective:

These are the most used functions on the entire website, so it is critical to know how well these functions are doing and to see the user's perspective on how to make it easier to use. Setting languages is one of the key functions of our website. Once the user sets the language they want to teach and learn, the backend finds the user's possible matches on the basis of their language preference so we need to test this functionality. Next we want to test our matching function, everything is done in the backend once a user sets a language. Users can see all the possible matches on the home page and then select other users to add them as their friends so we need to check this functionality to see how well it works and if it shows all the matches. We would like to understand what needs some changing to be more usable. After users get matched they can start chatting with their friends. Our website is for learning and teaching the language and it is done via chatting which makes this function necessary for being tested. Profile page is one of the other functionality we want to test. This is the page where you can see users details. It is important for this page to be tested since users can make all the changes like language, location, icons, description about them and add their post in this page. Posting is another functionality used in our website, it can be found in the profile page of a user. This is the place where users write and post their daily updates. We need to make sure this functionality works perfectly for users so they are able to make posts easily.

## System Setup:

We set up our server by cloning the code base from GitHub, migrating databases, running the setup script and hooking up nginx & gunicorn to work with Django. We give our user a laptop with Chrome Browser version 81.0

## Starting Point:

The user must either register or log in to an account on our website which once completed is the starting point of the test. First they need to select their preferred language. Only when they choose the language it shows them their possible matches on the home screen. Setting the preference language and learning language is the first thing they need to do in our website to get possible matches, add them as friends, and chat. So, the starting point after signing up to our website is setting the languages.

## Intended Users:

Our intended users are everyday people. The beauty of it is that everyone is open to learning a new language at some point in their life and our website is there to help these people actually retain the knowledge they have learned by allowing them to speak the language to somebody who

understands fluently. This is the hardest part for people because a person can learn how to read, write and speak a language, but won't be able to have a conversation because they never practiced. Our website can be used by literally anyone.

### URL + What Is To Be Measured:

The url of our website is lingomingo.app

We would like to measure the usability of our website and how well the user understands how to navigate through the website since the user is the most important factor in how successful the website turns out to be. We are also trying to see what feedback the user has regarding the website. We would like our website to be enjoyable and truly serve its purpose by getting the users rating on our major functions. We would like to see the users perspective on our layout and how they like it. In the end we are just measuring how satisfied the user is with our website.

### Usability Task Description:

The Usability Task is for users to add language according to their preference and then check for the possible matches in the home page. After that, add the matches as their friend. Another test is to chat with the friend they added. Users need to make a post in their profile page for the next task and then need to go and edit their profile like add location or write description about them. The usability test will be completed when they get some matches after setting the language, adding them as friends and chatting with the friend they added. The user should be able to make the post and edit their profile successfully for the task to be completed.

### Usability test table

Test/Use Case	% Completed	Errors	Comments
Setting Languages	100%	No errors	Needs to change the formatting and font sizes, works properly but doesn't look good.
Matching	100%	No errors	Font is small for the match list and the title font is too big.
Post to Profile	100%	No errors	The button for add post is small, instead of separate plus sign button, clicking on add post should be easier.

Chat	100%	No errors	Chat works fine but it's just plain text. Both people chat on the same side which makes it kinda confusing. No spacing in the chat.
Edit Profile	90%	No errors	The add icon doesn't work. The edit profile page needs to be more organized.

## Questionnaire

Please put an x through the one you agree with the most.	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I understood how to use the website.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
It took me a long time to figure out how everything worked.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The matching process was easy to use and enjoyable.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Please leave any comments, feedback is greatly appreciated:

Everything works properly and it is great but the look of the website is not that great, should make some changes in fonts and design of the website. Some of the stuff looks all clustered, so making it better will make the website great.

# QA Test Plan

## Non-Function Test 1

### Test Objective:

We are testing how things would look and work under different browsers to ensure good user experience

### HW and SW setup:

HW: LingoMingo Development Server on AWS

SW: Chrome, FireFox, Opera, IOS Chrome and LingoMingo code base on development branch as of 05/09/2020

### Features to be tested:

LingoMingo's compatibility on different browsers

### QA Test Plan:

Load all pages of LingoMingo on different browsers and see if it renders the way we want it to/ is functional

Test #	Test Title	Test Description	Test Input	Expected Output	Actual Output
1.1	Compatibility	Site Content Render Test	Chrome 81.0.4 64 bits	Fully Compatible	PASS
1.2	Compatibility	Site Content Render Test	Firefox Version 76.0.1 64 bits	Fully Compatible	PASS
1.3	Compatibility	Site Content Render Test	Mobile Chrome Version 81 on iOS 13.3	Partially Compatible	FAILED
1.4	Compatibility	Site Content Render Test	Opera Version 68.0.3 64 bits	Fully Compatible	PASS

## Non-Function Test 2

### Test Objective:

We are testing LingoMingo's load speed for both the pages to ensure quick and snappy experience.

### HW and SW setup

HW: LingoMingo Development Server on AWS

SW: Chrome and LingoMingo code base on development branch as of 05/09/2020

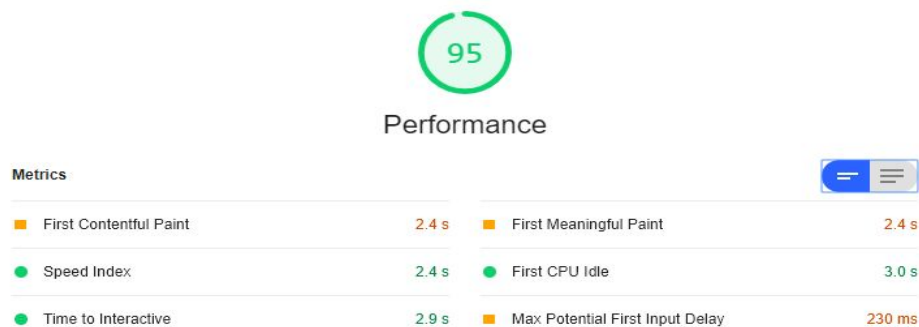
### Features to be tested:

LingoMingo's pages load speed on browsers,

### QA Test Plan:

We use Chrome's built-in auditing tool [Lighthouse](#) to perform page load speed test

Test #	Test Title	Test Description	Test Input	Expected Output	Actual Output
2.1	Performance	Load Speed	/ (logged in and not logged in)	<4.0 seconds for load time	2.9 s PASS
2.2	Performance	Load Speed	/profile	<4.0 seconds for load time	50s FAILED (with 50k posts, need to implement pagination )
2.3	Performance	Load Speed	/settings	<4.0 seconds for load time	2.9s PASS





## Non-Function Test 3

### Test Objective:

We are testing LingoMingo's database queries to ensure fast searching of data.

### HW and SW setup:

HW: LingoMingo Development Server on AWS

SW: Chrome and LingoMingo code base on development branch as of 05/09/2020

### Features to be tested:

LingoMingo's retrieval speed from database.

### QA Test Plan:

We wrote scripts to test query time for 50k post entries on our database

Test #	Test Title	Test Description	Test Input	Expected Output	Actual Output
3.1	Performance	Query Time	Search Post by ID	<1 seconds for load time	0.001977s PASS
3.2	Performance	Query Time	Search Post by Content	<1 seconds for load time	0.001975s PASS
3.3	Performance	Query Time	Search Post by Profile	<1 seconds for load time	0.002739s PASS

## Non-Function Test 4

### Test Objective:

We are testing image uploading and to make sure image format isn't a concern.

### HW and SW setup:

HW: LingoMingo Development Server on AWS

SW: Chrome and LingoMingo code base on development branch as of 05/09/2020

### Features to be tested:

Images shall be uploaded in jpg, jpeg, or png formats.

### QA Test Plan:

We will upload different formats of files in extreme conditions: image size, file size, non image files

Test #	Test Title	Test Description	Test Input	Expected Output	Actual Output
4.1	Data Integrity	Image Upload Test	LARGE IMAGE SIZE PNG/JPG	5k by 5k pixels Pass	PASS
4.2	Data Integrity	Image Upload Test	LARGE FILE SIZE PNG/JPG	Pass if <100MB	PASS
4.3	Data Integrity	Image Upload Test	NON IMG FILES	Does not get uploaded	PASS

## Non-Function Test 5

### Test Objective:

We are testing our site's ability to rescale in size in different browsers.

### HW and SW setup:

HW: LingoMingo Development Server on AWS

SW: Chrome and LingoMingo code base on development branch as of 05/09/2020

### Features to be tested:

Elements in screen shall be resized automatically without user interaction when being loaded in all the different platforms supported by the site

### QA Test Plan:

We will stretch and resize LingoMingo's pages and see if everything scales automatically.

Test #	Test Title	Test Description	Test Input	Expected Output	Actual Output
5.1	Look and Feel Standards	Element Resize Test	Chrome	Resizeable, PASS	PASS
5.2	Look and Feel Standards	Element Resize Test	FireFox	Resizeable, PASS	PASS
5.3	Look and Feel Standards	Element Resize Test	Opera	Resizeable, PASS	PASS

# Code Review

Our style of coding is eXtreme Programming (XP). This is the style that works best for us because we are such a small group of people working on the project. As a group we focus on simplicity first and then when things work we try to improve each aspect of our application. We also spend a lot of time working together and coding together on zoom calls and discord chats. This has helped with front end and back end working together.

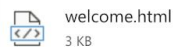
## From Arjun Bista to Sawara Bhattra



Arjun Bista

Sun 5/10/2020 2:16 PM

To: Sawara Bhattra



welcome.html

3 KB

Hello Sawara,

I looked on your welcome page, it looks good. Though I found some space for improvements for the frontend part. I have given some feedback and hope these comments are useful to you.

Regards,

Arjun Bista

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <!-- code review for welcome page
4     - Code looks good but still there is space for improvement.
5     | font is small, you can increase font size
6     - Also, I would like you to change background color or you
7     | can use same colors instead of two different.
8     -I would also recommend you to use bootstrap's version 4.
9   -->
10
11 <head>
12   <meta charset="utf-8">
13   <meta name="viewport" content="width=device-width, initial-scale=1">
14   <title>LingoMingo</title>
15   <meta name="description" content="Free Bootstrap Theme by BootstrapMade.com">
16   <meta name="keywords" content="free website templates, free bootstrap themes, free template, free bootstrap, fre
17
18   <link rel="stylesheet" type="text/css" href="https://fonts.googleapis.com/css?family=Open+Sans|Candal|Alegreya+S
19   <link rel="stylesheet" type="text/css" href=" ../css/font-awesome.min.css">
20   <link rel="stylesheet" type="text/css" href=" ../css/bootstrap.min.welcome.css">
```

## From Arjun Bista to Cassie Sherman

### peer code reveiw



Arjun Bista

Sun 5/10/2020 2:59 PM

To: cassiesherman630@gmail.com



setting.html

11 KB

Hello Cassie,

Your code looks good and has followed all the styles that we have discussed. You have done a great job. I don't see much room for the improvement, though I found a small glitch need to fix. Hope my comments will help you.

Regards,

Arjun Bista

```
login.html  <> welcome.html  <> setting.html  <> registration.html  <> registration1.html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <!-- peer code review for setting page
5  - Wow, everythings looks great. It is simple, neat, clean,
6  |   easy to use and well designed.
7  - I found that you need to work on email notification botton,
8  |   one of the botton is not working.  -->
9
10 <head>
11   <meta charset="utf-8">
12   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
13   <link rel="stylesheet" href="css/settings.css">
14
15   <script src="jquery.js"></script>
16   <script>
17     $(function(){
18       $("#includedContent").load("nav.html");
```

```
<> login.html <> welcome.html <> setting.html <> profile.html • <> registration.html <> registration1.html
1 <!DOCTYPE html>
2 <html lang="en">
3   <!-- Peer code review for profile page
4     Your Code follows our coding style, everythings looks so great! Im impressed with your works.
5     Code is easier to read because of proper indentation, though it would be better if you would provide
6     some comments to explain the code but not necessary -->
7
8 <head>
9   <meta charset="UTF-8">
10
11   <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootst
12   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
13   <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
14   <link rel="stylesheet" href="../css/profile.css">
15   <!-- Include the above in your HEAD tag ----->
16   <script src="jquery.js"></script>
17   <script>
18     $(function(){
19       $("#includedContent").load("nav.html");
20     });
21   </script>
22 </head>
23 <body>
24 <div id="includedContent"></div>
25 <div class="container emp-profile">
26   <form method="post">
27     <div class="row">
28       <div class="col-md-4">
29         <div class="profile-img">
```

# Best Practices For Security

In LingoMingo, the most important assets are our user's information, such as passwords, chat logs, pictures in their posts. We encrypt our user's password using Django's authentication system which by default uses PBKDF2 algorithm with SHA256 hash with salt. In the following image, our password field value in the database is stored as `<algorithm>$<iterations>$<salt>$<hash>`

(<https://docs.djangoproject.com/en/3.0/topics/auth/passwords/>)

The screenshot shows a Django password form. The 'Username' field contains '213edu'. Below it, a note says 'Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.' The 'Password' field is empty. Below it, the database storage format is shown: 'algorithm: pbkdf2\_sha256 iterations: 180000 salt: 3vB2jt\*\*\*\*\* hash: fWax61\*\*\*\*\*'. A note below that says 'Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.'

Users have the option to change their personal profile, create posts containing photos/text, and search for users. We do validation on every POST request that the user passes in.

```
if request.POST.get('search'):
    query = request.POST.get('search')
    search_results = get_profile_model().filter(
        Q(user__first_name__icontains=query) | Q(user__last_name__icontains=query))
    if search_results.count() == 0:
        messages.error(request, 'no result for the search query')
    context.update({'search_results': search_results})
```

For example, when users pass in a search query for searching other's profiles by name, we use Django's query method to get the result as QuerySet then returns the QuerySet to our template rendering that will only output the name of the users, Preventing SQL injections. As for pictures and chat, we use Django's built-in user authentication verification to ensure only registered users with sufficient permission can view the requested files and logs.

```
{% if search_results %}
<h5>Results:</h5>
<ul>
    {% for profile in search_results %}
        <li>
            <a href="{% url 'profile' profile.uuid %}">{{ profile.user.first_name }} {{
profile.user.last_name }}</a>
        </li>
    {% endfor %}
</ul>
{% endif %}
```

# Adherence To Original Non-Functional Specs

## Security

Login shall be required to utilize website and learn a language	DONE
User's shall verify their emails when registering an account	N/A
User's shall be able to set a display name different than their email	DONE
User's emails shall not be displayed by default	DONE
Passwords shall be encrypted before storing in the database	DONE
Content uploaded by users shall be audited by the administrator	ON TRACK
User's payment information shall be encrypted	Issue, we don't have premium accounts yet
This site shall not accept any third party cookies	N/A

## Audit

1. New registrations shall be audited by the administrator	N/A
2. New registrations shall be approved by the administrator	N/A
3. Users shall not be able to login to administrator accounts	DONE



## Performance

1. The site loading time shall be less than 4 seconds for all screens	ON TRACK
2. Application shall be able to retrieve information from the database and react in a timely manner.	ON TRACK
3. The site shall handle requests asynchronously following a REST format	N/A

## Capacity

1. The total data storage for the site shall not exceed 80% of the server's capacity for this site	ON TRACK
2. The website shall be capable of handling at least 50 users	ON TRACK
3. The website shall be scalable, so that new features can be added easily	ON TRACK

## Reliability

1. Downtime for maintenance shall be less than 3 hours per month	ON TRACK
2. Downtime for maintenance shall not affect the site's main functionality	ON TRACK
3. In all cases, users shall be informed of downtime for maintenance, either via an announcement on the main page, or e-mail	ON TRACK

## Recovery

1. In case of a total site failure, the whole site shall be shut down for revision.	ON TRACK
2. If the site is broken, the mean time to recovery shall not exceed one day.	ON TRACK
3. User data is the most valuable aspect and priority will be placed on recovering such data in case of total failure.	ON TRACK

## Data Integrity

1. Database tables shall be backed up weekly	ON TRACK
2. Administrator shall be able to execute a recovery if needed	ON TRACK
3. Image sizes shall be restricted to at most 1 megabyte	ON TRACK
4. Images shall be uploaded in jpg, jpeg, or png formats	ON TRACK
5. Images will be saved on Amazon's s3 storage server	ON TRACK
6. URLs to image will be stored on the database	ON TRACK

## Compatibility

1. The site shall be compatible with the last version of the Safari browser version 11.1.2	ON TRACK
2. The site shall be compatible with the last version of the Firefox browser version 70	ON TRACK

3. The site shall be compatible with the last version of the Chrome browser version 80	ON TRACK
4. Third party applications shall not be able to modify any content that may affect the site compatibility	ON TRACK
5. Content should be able to be ignored by most popular ad-block services.	ON TRACK
6. The site shall be able to account for any other compatibility issues created as a result of browser updates in the future	ON TRACK
7. The site should be compatible to escalate to new databases	ON TRACK

## Conformance with Coding Standards

1. Architecture and design standards shall meet all the requirements listed under the High-level system architecture and technologies used section of this document	ON TRACK
2. Design pattern is to be strictly enforced with all aspects of the site.	ON TRACK
3. Appropriate documentation must be created for all code that is individually written for future maintenance.	ON TRACK
4. Production code shall not have any log or output to the console.	ON TRACK
5. All errors must not halt the web application without appropriate error handling.	ON TRACK
6. Only working code that meets all code standards shall be submitted to the main branch of the project repository	DONE
7. Code shall be thoroughly tested and debugged before being considered working code	DONE
8. All internal errors and exceptions encountered when writing or modifying code shall be stored in a log	ON TRACK

9. Any error that can affect the site's functionality shall be reported to the user	ON TRACK
10. Errors shall be handled in a way that does not affect site functionality	ON TRACK
11. The whole production cycle of the site shall be finished at least one week before the delivery date	ON TRACK
12. The site shall be tested and debugged as a whole product at least one week before the delivery date	ON TRACK
13. The site shall not be launched without all priority one features finished and working	ON TRACK
14. All major changes to the application shall be discussed by the team and communicated to the class CTO.	ON TRACK

## Look and Feel Standards

1. The application and it's layouts shall look professional	ON TRACK
2. The site shall be simple, so that it is usable to a wide range of users, and all previously mentioned parties	ON TRACK
3. Targeted users will be the main priority for ensuring usability and readability.	ON TRACK
4. Elements on screen shall meet the compatibility standards of all supported browsers	ON TRACK
5. Elements on screen shall meet the compatibility standards of all supported browsers on mobile devices	ON TRACK
6. Elements on screen shall be aesthetically pleasing	ON TRACK
7. The site shall be able to work correctly without mouse interaction	ON TRACK

8. The site shall be able to work correctly without keyboard interaction	DONE
9. Elements in screen shall be resized automatically without user interaction when being loaded in all the different platforms supported by the site	DONE
10. Application's user interface shall make it easy for users to find what they are looking for.	ON TRACK

## Scalability

1. The CPU instance and storage capacity shall be updated to be able to handle a large amount of users if needed	ON TRACK
2. It should be able to add extra resources and also keep the structure intact.	ON TRACK
3. Individual users should be able to use the app efficiently and properly even when the number of users keeps increasing.	ON TRACK

## Web Site Policies

1. A link to the policies of this site shall be always visible in its settings page to be accessible by all the parties	ON TRACK
2. The site will not store any payment information.	NOT AVAILABLE
3. Users payment information shall be kept confidential and secure	NOT AVAILABLE
4. The website shall allow users to register an account.	DONE
5. Email verification shall be implemented upon registration.	ON TRACK

6. User's shall agree to the application's privacy policy before using the product.	DONE
---	------