Swe574 Customer Milestone 2 Deliverables

Milestone Review

- The requirements addressed in this milestone

R.2.1.01	Each user should be able to see in their activity feed when a user in their follow list posts a story.	Activity Feed	-
R.2.1.02	Users should be able to see the stories that take place in a nearby location in chronological order.	Search	Location
R.2.1.03	Users should be able to increase or decrease the story selection areas by adding distance information when searching based on location.	Search	Location
R.2.1.04	Users should be able to add a date-time filter to the stories.	Time	
R.2.1.05	Users should be able to se arch stories based on location and distance.	Search	
R.2.1.06	Each user should be able to see in their recommended feed the stories related to the following users' stories labels, locations, dates.	Recommen dation	-
R.2.1.07	Users can save stories and view them later.	Save	Story Detail
R.2.1.06	Each user should be able to see in their recommended feed the stories related to the following users' stories labels, locations, dates.	Recommen dation	-

Deliverables

Req Id	Requirements Addressed in This Milestone			Mobile Progress	Web Progress
R.2.1.01	Each user should be able to see in their activity feed when a user in their follow list posts a story.	Activity Feed	-	Completed	Completed
R.2.1.02	Users should be able to see the stories that take place in a nearby location in chronological order.	Search	Location	Completed	Completed
R.2.1.03	Users should be able to increase or decrease the story selection areas by adding distance information when searching based on location.	Search	Location	Completed	Completed
R.2.1.04	Users should be able to add a date-time filter to the stories.	Time		Completed	Completed
R.2.1.05	Users should be able to search stories based on location and distance.	Search		Completed	Completed
R.2.1.06	Each user should be able to see in their recommended feed the stories related to the following users' stories labels, locations, dates.	Recommendation	-	In progress (enhancements planned)	In progress (enhancements planned)
R.2.1.07	Users can save stories and view them later.	Save	Story Detail	Completed	Completed

Legend: Not started, In progress, or Completed (Completed means all of the following: the feature is implemented, tested, documented, and deployed).

Mobile Application Deliverables:

- Activity feed where users can view stories based on following users.(Completed)
- Recent Feed where users can view stories that have been uploaded in the last 7 days. (Completed)
- Add Story with multiple locations, location region selection, time resolutions, and photo upload. (Completed)
- Follow other users to interact. (Completed)
- Like stories and view them under ‘Liked Stories.’(Completed)
- Save stores and view them under ‘Save Stories’. (Completed)
- View the user’s own stories under ‘My Stories’.(Completed)
- Nearby stories where user view stories based on their location.(Completed)
- Choose the radius in nearby stories to view closer or farther away stories. (Completed)
- View story details including story location, title, the story itself, and their comments.
- Add comments under stories. (Completed)
- Update user profile, biography, and profile picture. (Completed)
- Timeline search where users can search stories based on their date, location, decade, season, title, and tags. The result returns the intersection of the search query. In the timeline search, choose the location from a map.(Completed)
- The search page is where users can search stories based on their date, location, decade, season, title, and tags. The result returns the combination of the search query. In the timeline search, choose the location from a map.(Completed)
- Recommended stories where users can view recommended stories based on their activities in the app, such as following other users, liking stories, etc.(Completed)

Web Application Deliverables:

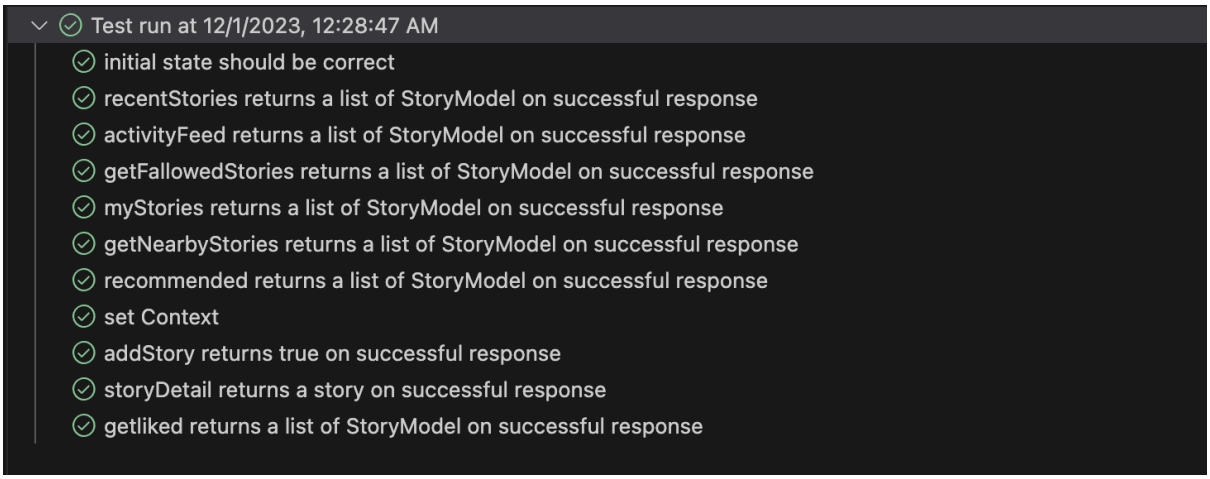
- Time resolutions fixes from previous customer feedback (Completed)
- Tag Component Enhancements (Ongoing)
- Tag search (Completed)
- Map drawing is integrated (Completed)
- User’s location feature is added now (Completed)
- Recommendation page (Completed)
- timeline page (Completed)
- Editing a story is available (Completed)
- New Message Component Integration (Completed)
- Navigation Bar is enhanced (Completed)
- QR Redirecting is added to the main page (Completed)
- Bug fixes about null values (Completed)
- Bug fixes about value requirements (Completed)
- Bug fixes about time values (Completed)
- Enhancements about UI forms (Ongoing)

Testing

- The general test plan for the project, which describes your product's testing strategy (e.g., unit testing, integration testing, mock data, etc.).

Mobile App Testing:

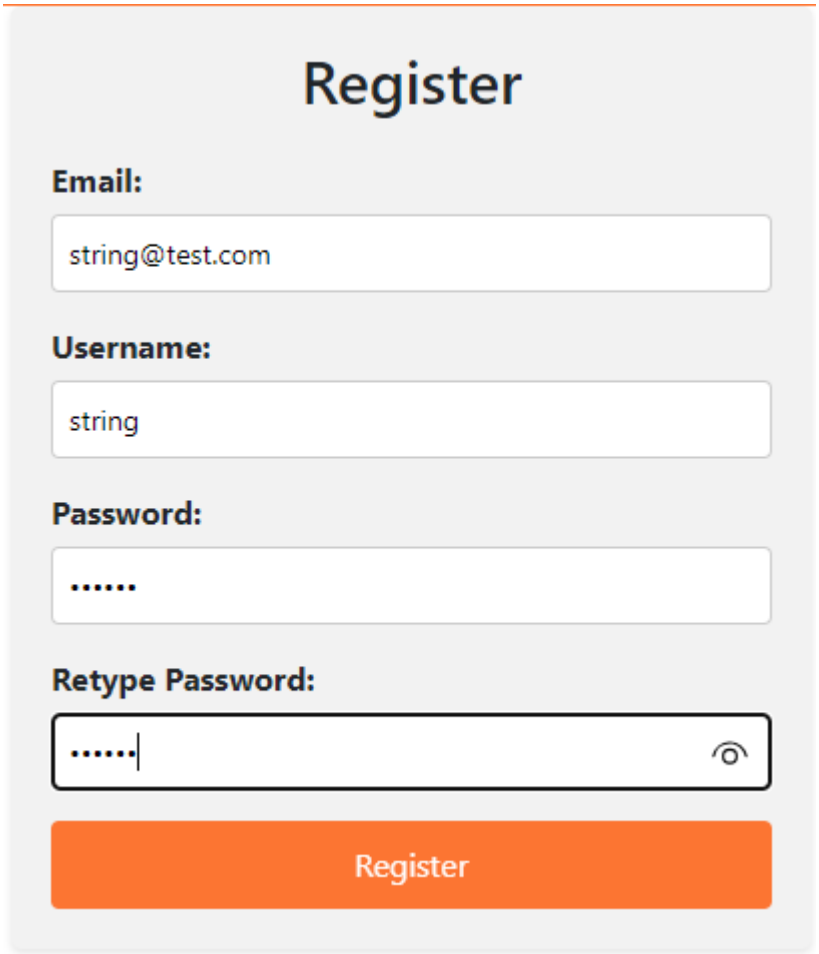
UAT for mobile app testing is conducted through Firebase releases. We test the mobile app on our android phones and check the features functionalities. In addition, unit tests are written for the story related functions. For the moment, unit tests are for getting recent stories, activity feed, followed user stories, my stories, nearby stories, recommended stories ,liked stories, story detail, and sending add story requests.



Web App Testing:

Register Page:

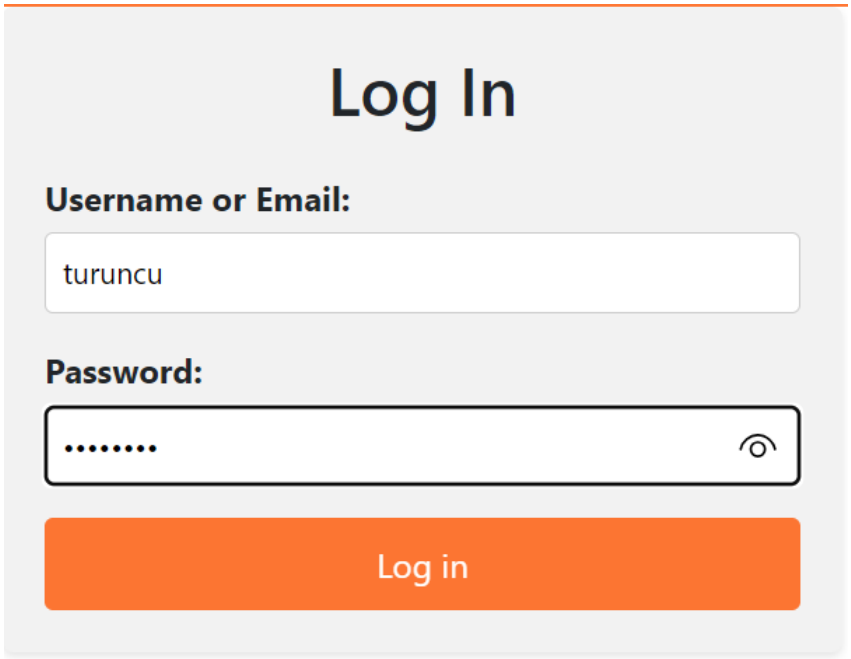
- In this page this is the default value that we test whether the register is working.



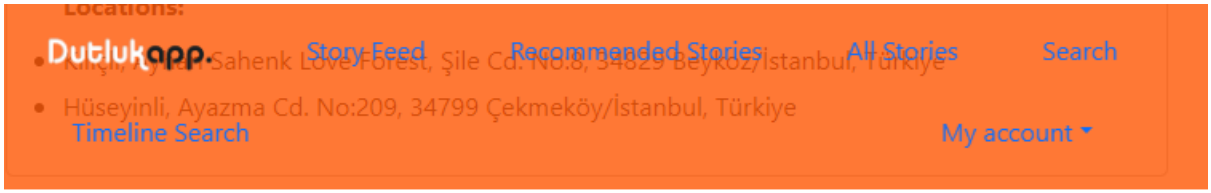
-

Login Page:

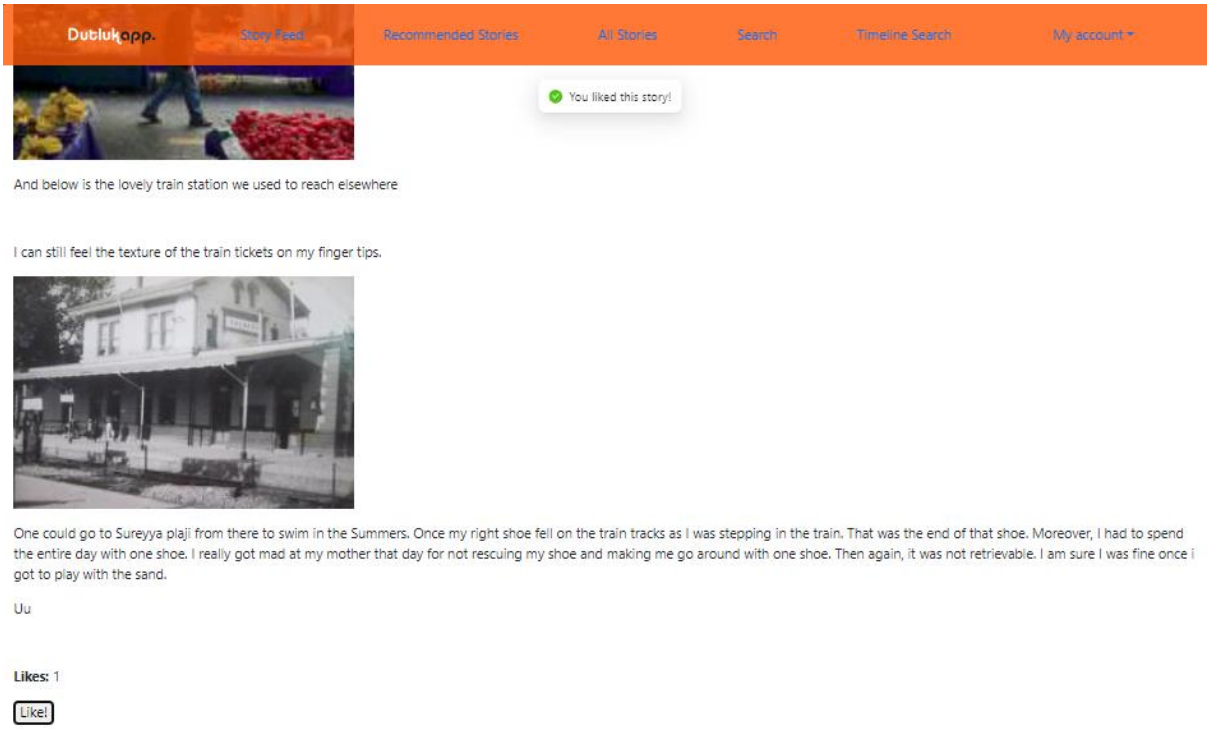
- In this page this is the default value that we test whether the login is working.



All Stories Page:

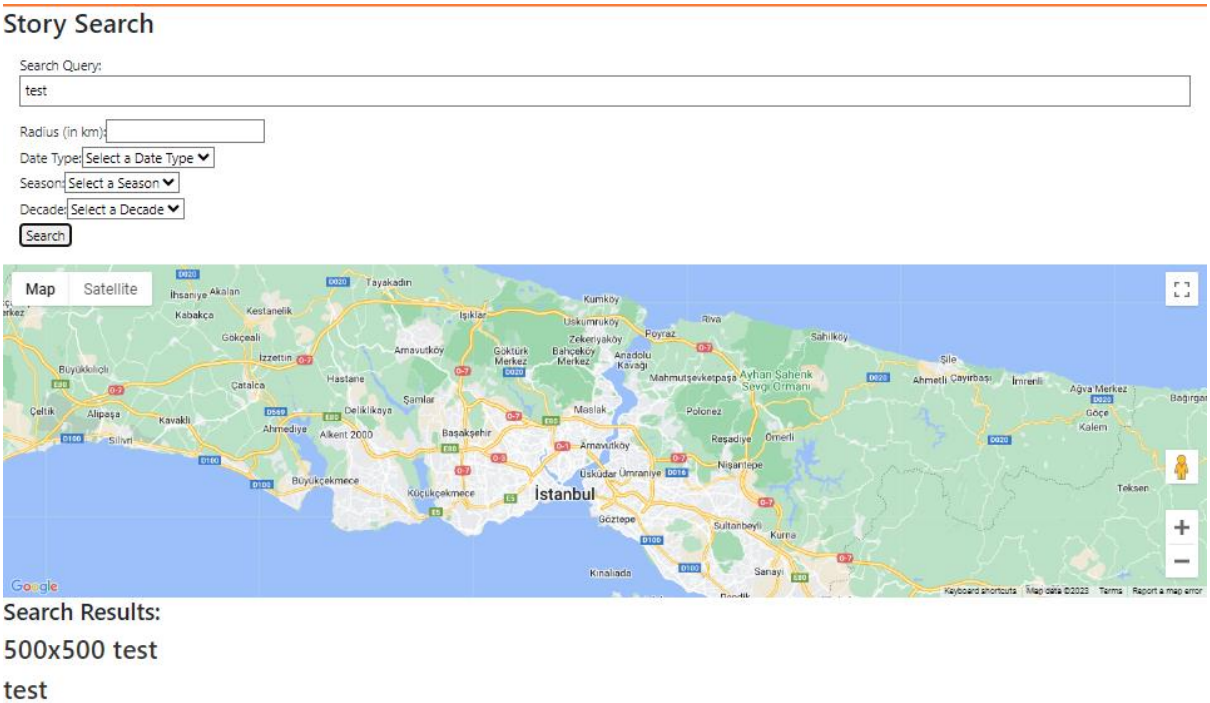


Story Details:



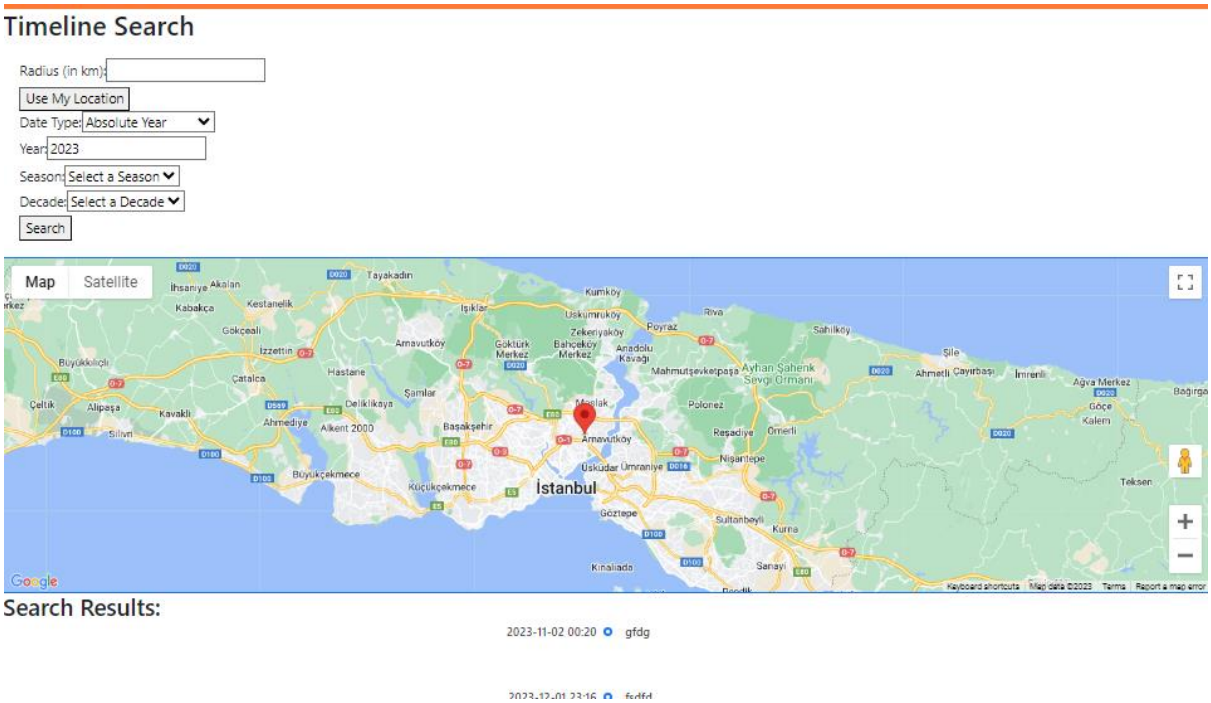
Search Page:

- Search by query, time resolution and location is tested.

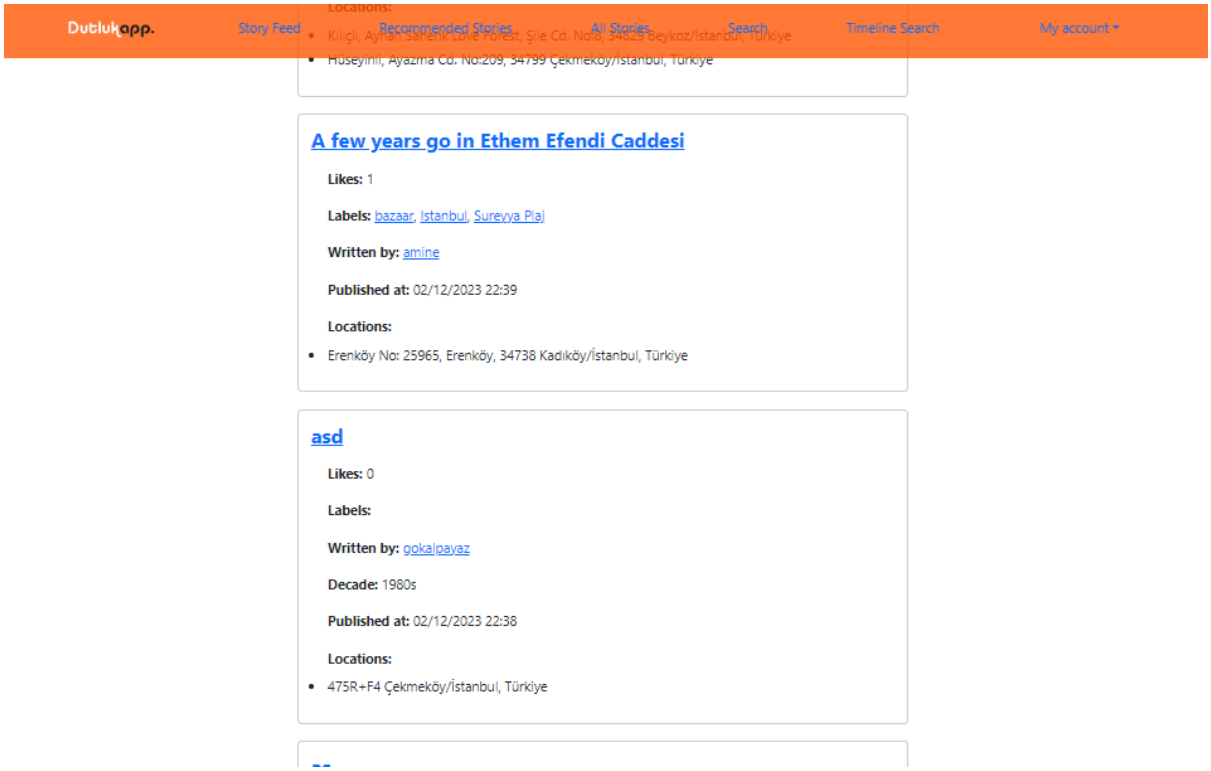


Timeline Search Page:

- Location and time resolution is done here.

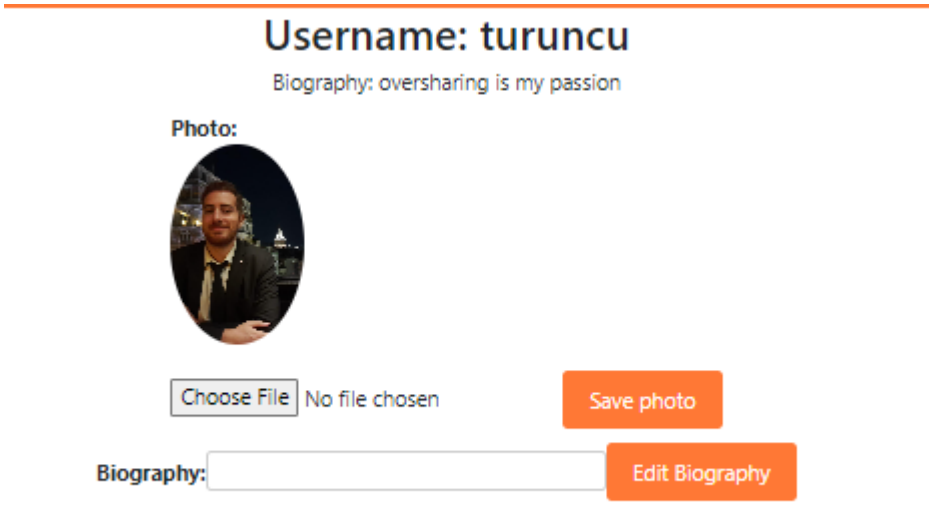


Recommended Stories Page:



My Profile Page:

- Here we update biography and profile picture.



Add Story Page:

- One of the default values to check add story functionality.

Locations:

VİALAND AVM, Yeşilpınar, 34065 Eyüpsultan/İstanbul, Türkiye Remove

Title: my fun time at vialand

Labels:(comma separated) like tatilya,interesting,enjoyable

Text:

Normal

I was at vialand hoping it would be something similar to Tatilya but it only made me sad.

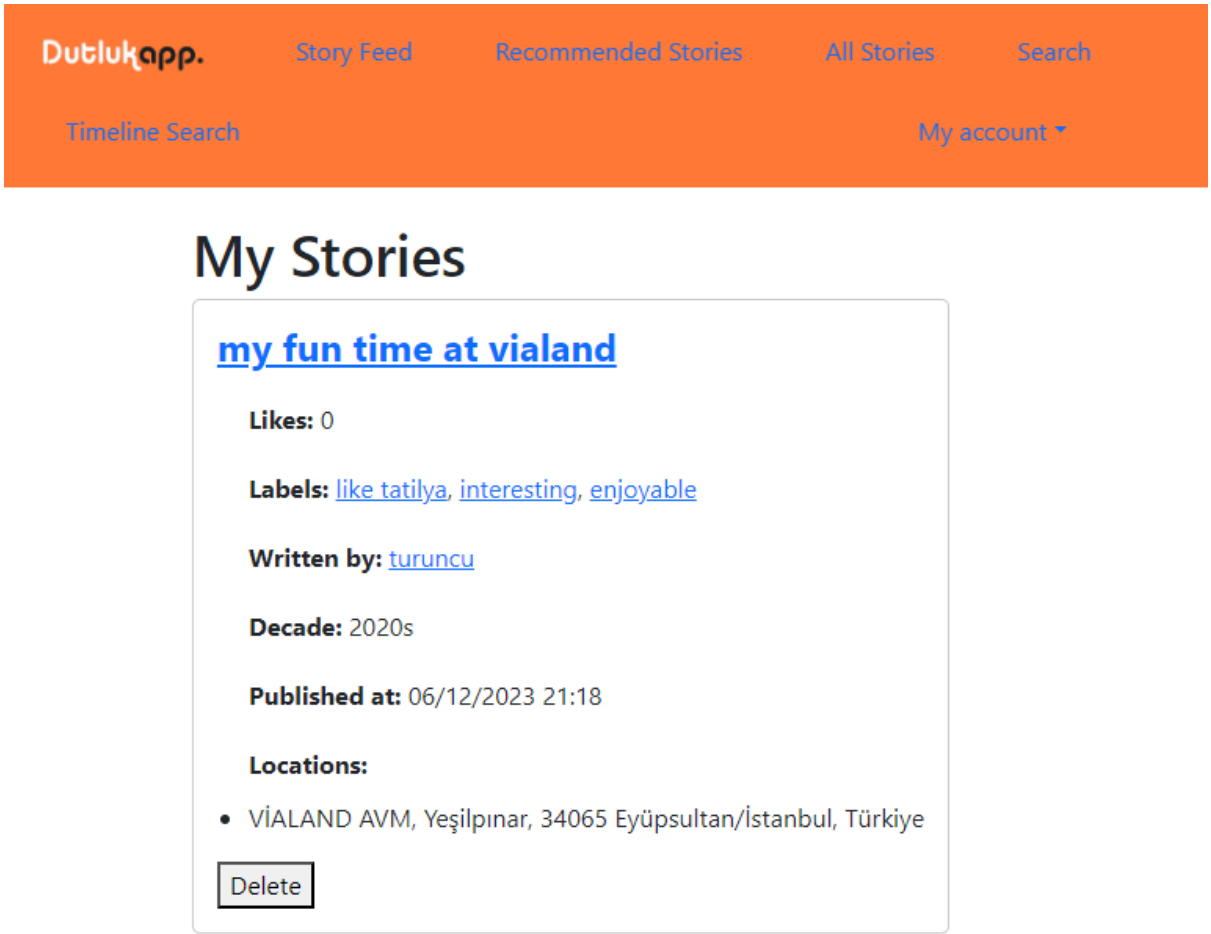
Start Date and Time: 10/27/2019 11:00 AM

End Date and Time:

Season: Select Season

My Stories Page:

- Delete function is tested here.

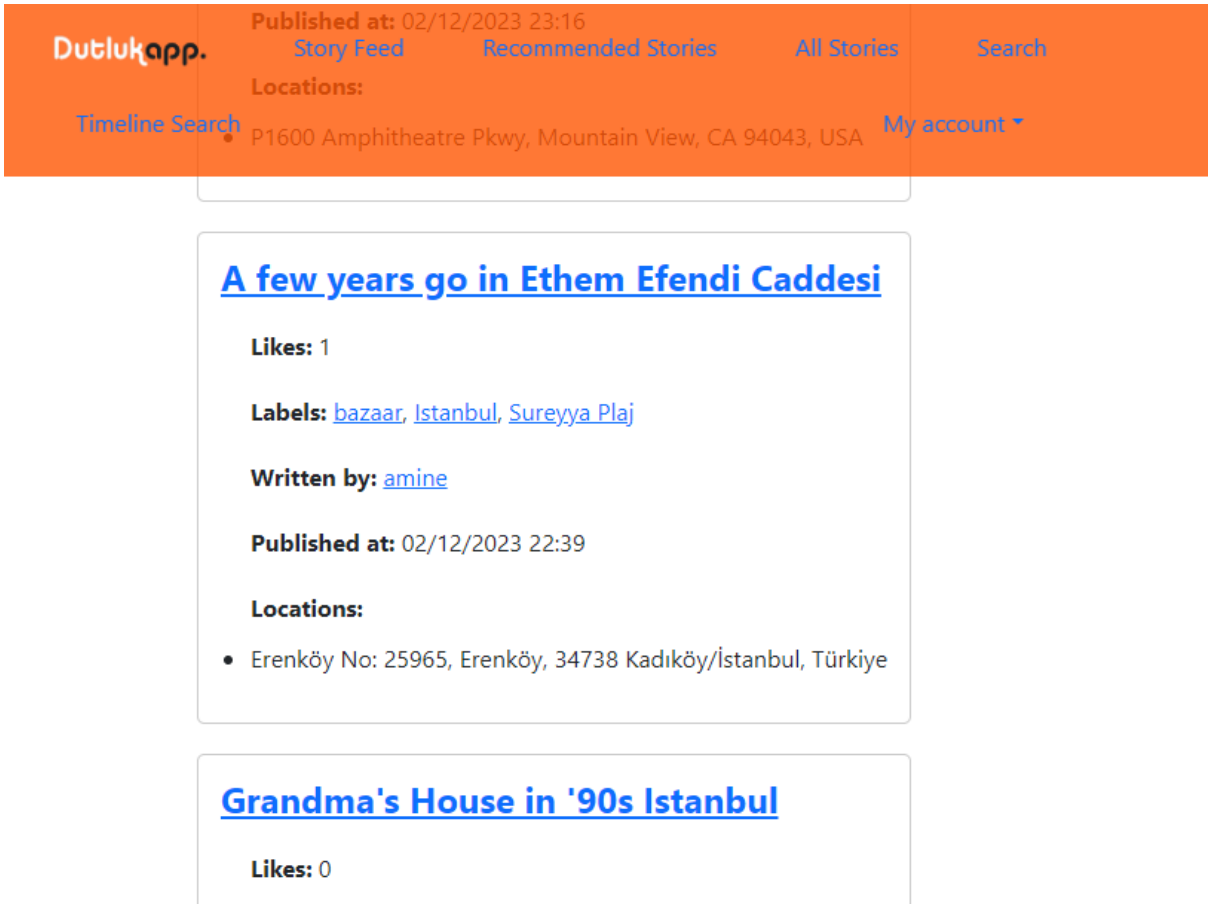


Other User Profile:

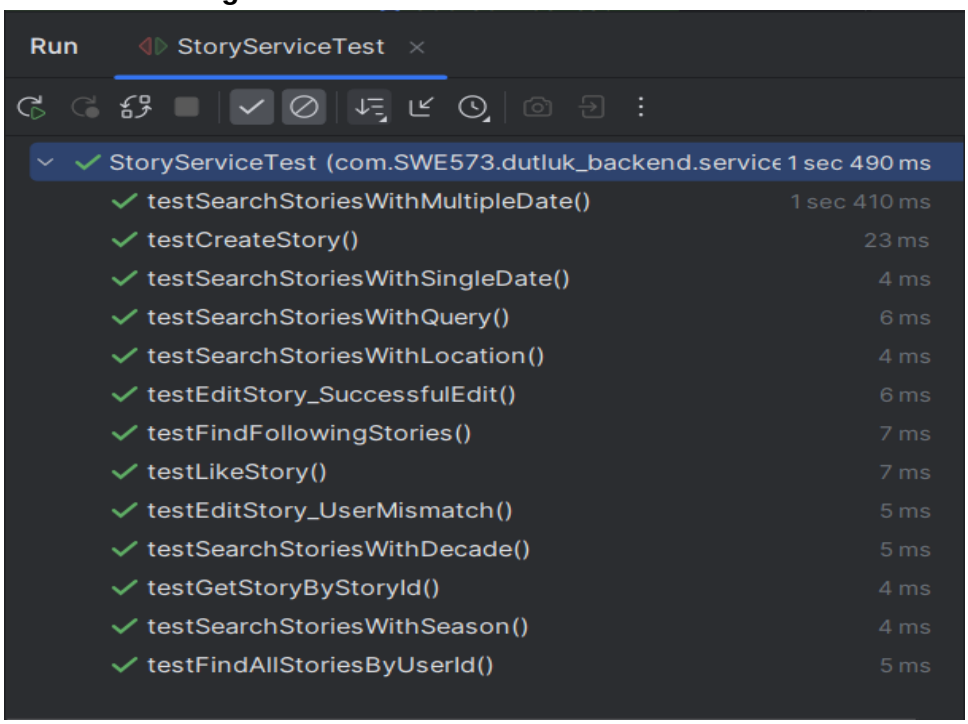
- Follow/unfollow is tested here.

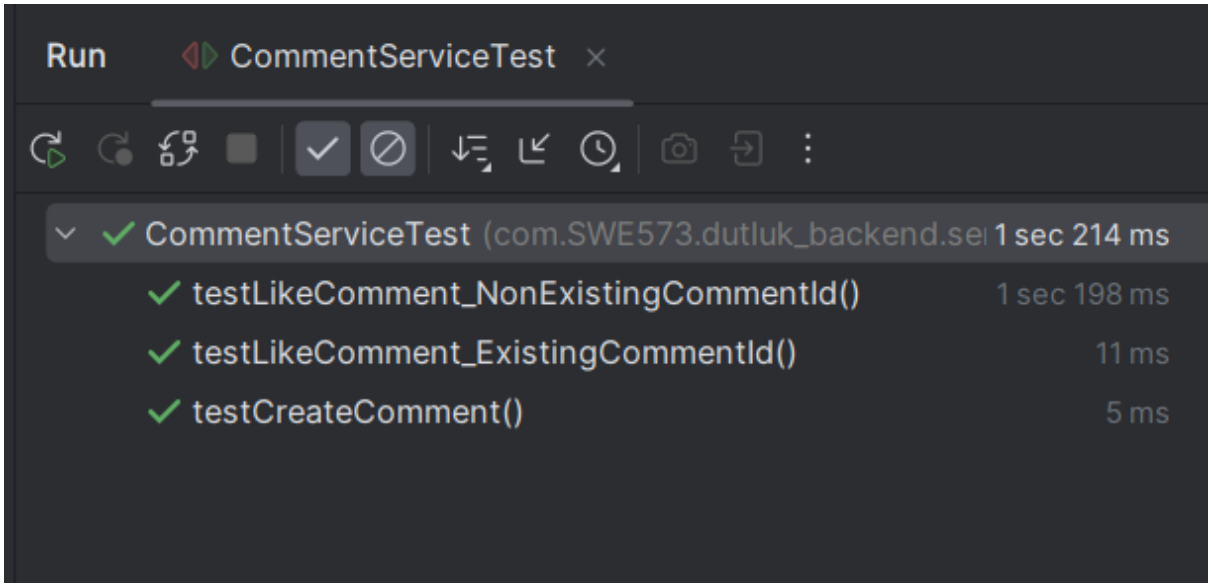
Followed User Stories:

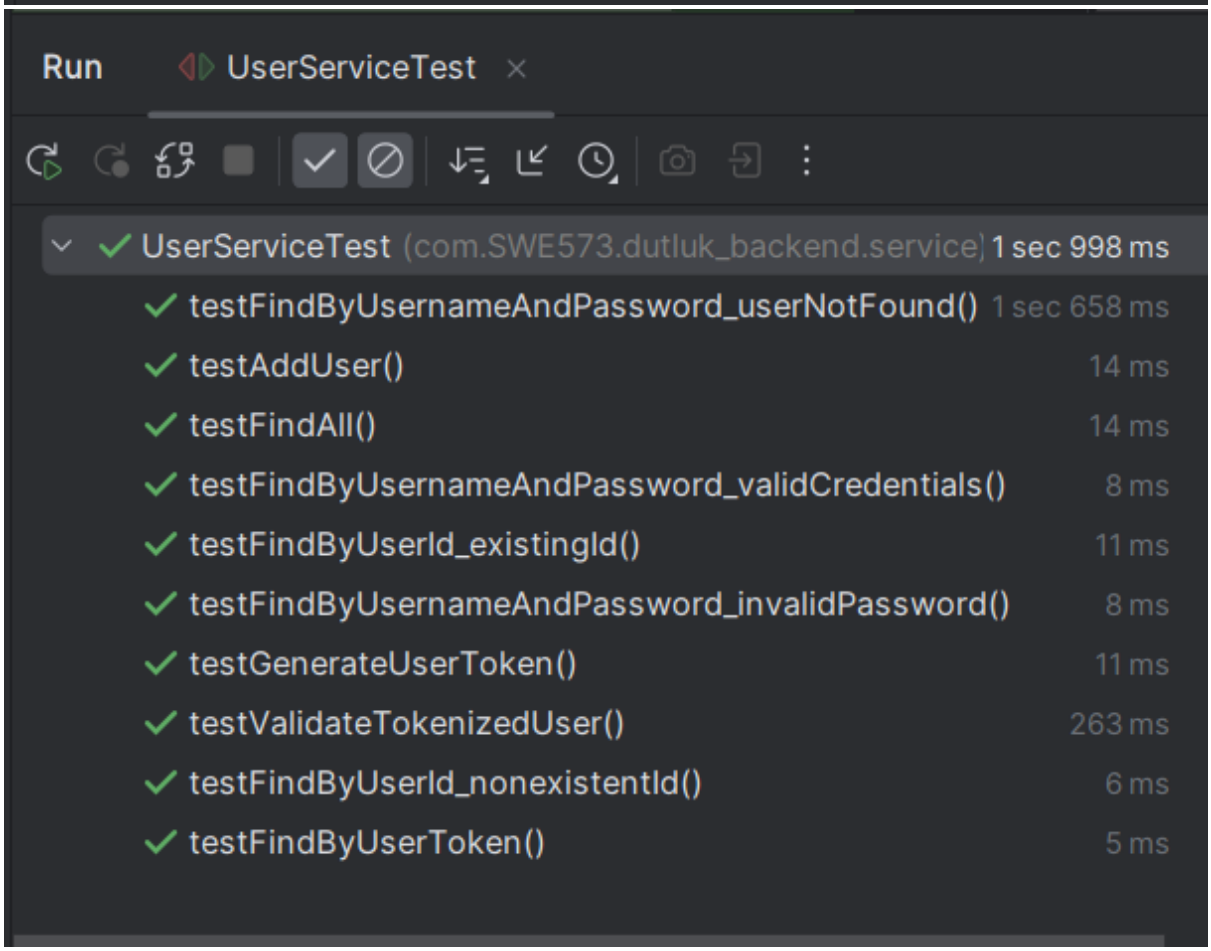
- Following/Unfollowing functionality tested here.



Backend Testing:




Run
CommentServiceTest
CommentServiceTest (com.SWE573.dutluk_backend.se 1 sec 214 ms
testLikeComment_NonExistingCommentId() 1 sec 198 ms
testLikeComment_ExistingCommentId() 11 ms
testCreateComment() 5 ms


Run
UserServiceTest
UserServiceTest (com.SWE573.dutluk_backend.service) 1 sec 998 ms
testFindByUsernameAndPassword_userNotFound() 1 sec 658 ms
testAddUser() 14 ms
testFindAll() 14 ms
testFindByUsernameAndPassword_validCredentials() 8 ms
testFindByUserId_existingId() 11 ms
testFindByUsernameAndPassword_invalidPassword() 8 ms
testGenerateUserToken() 11 ms
testValidateTokenizedUser() 263 ms
testFindByUserId_nonexistentId() 6 ms
testFindByUserToken() 5 ms

Planning and Team Process

For the mobile version, we introduced several new features. These include the Recent Stories, Activity Feed, Recommended Page, Nearby Page, Timeline Page, and Search Page. We also added the ability to like and save stories, use map drawing tools, search by tags, and update the user's profile and profile picture. Additionally, we implemented a feature for adding comments to stories. These changes have made the mobile app more interactive and user-friendly.

On the web platform, we enhanced the Tag Component, integrated Tag Search and Map Drawing, and utilized the user’s location feature. We developed the Recommendation and Timeline Page, implemented the ability to edit stories, and enhanced the NavBar. We also introduced QR Redirecting and fixed issues with null values and required values. Lastly, we improved the process of adding stories. These enhancements have made the web platform more robust and efficient.

Looking ahead, we plan to continue improving both platforms. We aim to streamline the user interface and make the platforms more intuitive. These changes are expected to make our development process more efficient and enhance the overall user experience. By constantly updating and refining our features, we hope to keep our platforms up-to-date with the latest technology trends and user needs.

Planned Developments for Final Presentation:

- Notification feature
- Better exception handling
- Karadut improvement on user taste analysis and implementation
- Saved Stories implementation on Web App
- Further UI/UX fixes on all platforms

Evaluation

The customer has provided valuable feedback on how to improve the story-making tool. They pointed out an issue with the date-time format. When a user writes a story, choosing dates and times can be complex because it involves elements like months and seasons. The customer suggests that the tool should be enhanced to better understand and integrate these aspects. This improvement would help avoid confusion between months and seasons, ultimately leading to a smoother user experience.

Additionally, the feedback emphasizes the need for more flexibility in the user interface, especially when entering locations. If a user mistakenly inputs an incorrect location while creating their story, they should have a simple way to correct or delete this. This feature is important to allow users to make changes easily.

Lastly, the customer highlights the importance of an editing feature. After creating a story, users might want to revise their work. Therefore, adding an option to edit the story is essential, giving users the ability to adjust their content as needed and enhancing their overall experience with the tool.

We have made significant progress in meeting most of our project requirements on at least one platform, with a notable achievement in the development of our recommendation engine. While the recommendation engine is currently operational and can be deployed, we recognize that it requires further enhancements to achieve optimal performance. In the meantime, it is sufficiently robust for current use. Concurrently, our team is dedicated to addressing and resolving frontend bugs. This ongoing effort is part of our commitment to refining the overall user experience and ensuring the stability and functionality of our platform. As we move forward, we will continue to focus on both the improvement of our recommendation engine and the resolution of any frontend issues, striving towards a seamless and efficient user interface.

In our project, we efficiently combined the use of Discord, WhatsApp, GitHub, Sonarcloud, CI/CD pipelines and Google Cloud Engine and Azure to manage our tasks and streamline

communication. For our weekly meetings and discussions about open issues, we utilized Discord. Its voice and screen sharing capabilities made it ideal for these interactive sessions. WhatsApp served as our go-to tool for day-to-day messaging and quick updates, offering easy access and convenience for all team members. GitHub played a crucial role in our code management, providing a platform for sharing, reviewing, and giving feedback on code changes. Lastly, we deployed our project on Google Cloud, which was a critical step in making our application accessible and functional.

Individual contributions

Individual Contribution:

○ **Member:** Aminenur Dağlargüler

○ **Responsibilities:** Mobile App Development

○ **Main contributions:** Mobile app development is completely done by me. I also assist the leading of the project, from opening the issues for bugs, feature requests. I also tested the development of the web app and mobile app.

○ **Code-related significant issues:**. Mobile add story feature, mobile recent stories page, activity feed page, nearby stories page, timeline search page, search page, like stories, save stories, update profile picture, update profile, leave comment on stories, follow people, my stories page, save stories page, liked stories page, story detail page with location map opening inside, add map tools to the map such as circle, polyline, polygon markers are implemented for the milestone 2 Here are all the issues assigned to me for milestone 2.

Assigned issues for Milestone 2

○ **Management-related significant issues:** Mobile app design are done by me. For the web app, I gave design ideas and best practice ideas to develop the applications. I also helped with the branch management of the project. I reported bug fixes or feature implementations for web app.

Here are all the issues that helped for the management of the project.

Closed Issues

○**Pull-requests:****Closed pull requests.** About 30 pull request either opened or merged by me**.** I have opened, closed and merged pull requests. Other than one pull request which was opened from a wrong branch and wanted a pull request to a wrong branch again, I didnt experience any conflict. For that pull request, I rejected the pull request and informed the author about his rejection. Generally, if I open a pull request, I send a branch demo to the reviewers and ask them to test it, then merge it if everything is alright. If I am a reviewer of a pull request, I test the feature from Swagger for a web application and if everything is okay after testing and reviewing the code I merge the pull request.

References of Contribution:

Contributions

Closed Issues

Created Issues

Closed Pull Requests

Lines of Code	257.300
Closed Issue Count	46
Open Issue Count	4
Pull Request	28
Commit Count	71

Individual Contribution:

○ **Member:** Hasan Deniz Doğan

○ **Responsibilities:** Product Owner, Tech Analyst, Q&A Tester, Backend & Frontend Development, Docker deployment, Rec. Engine(Karadut) integration

○ **Main contributions:** Assistance in leading the project, Development of new apis & services, optimizing data flow, Backend integration to Karadut and containerization. Saved stories implementation, pull request review on deployments, bug fixes etc.

○ **Code-related significant issues:** Timeline search implementation which works as an intersection instead of combination. End-to-end connection implementation with Karadut and Dutluk backend. Moving some logic to be checked on .env to have a loosely coupled system for uninterrupted testing. Branch testing on cloud servers to properly review pull requests. Image conversion from base64 to imgur links with img tags.

○ **Management-related significant issues:** Help positioning relevant triggers to properly implement features.Moving logic as much as possible to the backend to ensure mobile and web page optimization. Branch navigation to prevent conflicts. Local deployment assistance for different teams. Assisting all team members to solve personal and group management issues. Sharing my ideas on best practice uses. Action plan of implementing end-to-end connection with Karadut.

○ **Pull requests:** Closed Pull Requests Around 100 pull requests were either created, merged, reviewed and closed by me. Most of the conflicts were either caused by the wrong base branch being selected or 2 or more people working on the same files which caused unfortunate reworks. They were all solved by properly engaging means of communication.

○ **Additional information:** I also had to make several unexpected meeting calls to decide and resolve issues. Stated branch consolidation techniques and applied them along with the rest of the team.

References of Contribution:

Contributors

Closed Issues

Created Issues

Pull requests

Lines of Code	68.058
Closed Issue Count	42
Open Issue Count	2
Pull Request	26
Commit Count	221

Individual Contribution:

○ **Member:** Şakir Tevfik Özbilgin

○ **Responsibilities:** Fronted Development (Web, React)

○ **Main contributions:** Worked on Frontend Features especially on adding and editing forms, enhanced and updated these forms, refining and enhancements. Identified bugs and resolved them. Reviewed some pull requests. Worked on enhancing user experience, and increasing the code quality.

○ **Code-related significant issues:** I have worked on forms on web frontend. Identified some logical or minor errors on time resolution; made some basic changes on google maps about its inputs from the user, so that we are going to implement a better UI in the next milestone. Contributed to the enhancement of navigation bar. Updated the Web UI to become consistent with Mobile UI, changed some buttons to be more consistent in the Web UI, using Ant design.

○ **Management-related significant issues:** Attend to the weekly meetings, contributed to increase communication in web frontend side and helped to make decisions with about web frontend team. Contributed to the writing of meeting notes on some weeks, and helped deciding the way of milestone 2 presentation. Also reviewed some of issues.

○ **Pull requests:**

https://github.com/enshkn/BOUN-SWE-574-Fall-23-G2/pulls/assigned/TevfikOzbilgin

These pull requests are about add, and edit story forms, and search forms. They include enhancements about UI and google map radius inputs. Also include bug fixes and button changes.

○ **Additional information:**

References of Contribution:

Contributors

Closed Issues

Created Issues

Pull requests

Lines of Code	
Closed Issue Count	
Open Issue Count	
Pull Request	
Commit Count	

Individual Contribution:

○ **Responsibilities:** Enes Hakan IBIL

○ **Responsibilities:** Product Owner, Developer and Tester of Recommendation Engine (Karadut), Tech Analyst, Q&A Tester, Backend Development, Docker deployment, Rec. Engine(Karadut) integration.

○ **Main Contributions:** As of Customer Meeting 1, the recommendation engine (Karadut) plan was prepared; many alternatives used in recommendation systems were evaluated. Considering the structure of the stories used in our project, which can be considered as the backbone of the application, it was concluded that the most logical structure to be used for the recommendation engine is ***word embedding models.*** From this point on, from Milestone-1 to Milestone-2, my main focus was to stand up and integrate a working system using the word embedding model. As a member of the backend team, the development process of the Karadut project was a process that should run in ***parallel and separately from the application backend***, primarily because of the difference in the development environment used, while the application backend was developed in Java Spring Boot, Karadut would be developed in Python. This necessitated making us to determine some principles in advance and acting accordingly, because the module I developed would be added to the application at the end of the day and run as such. This led to some ***non-functional requirements in terms of quality such as compatibility, modularity, maintainability,*** all these requirements will be added to the project request page.

○ **Code Related Issues:** Currently Karadut has 6 endpoints, 23 functions and 3 class structures used in the functions of these endpoints were written in this development process. During the development process, 44 issues were opened and 57 commits were made. There is currently no open issue on me. As I mentioned in the previous section, Karadut is a module developed in parallel, so Backend-Integration meetings were held with Backend team member Hasan Deniz Dogan apart from weekly meetings.Apart from Karadut, the tasks belonging to other issues assigned to me as a backend team member were performed.

○ **Management Related Issues:** During the weekly meetings held throughout the development process, records were kept, planning was made, integration meetings were organized and wiki page edits were made on a rotational basis.

○ **Management Related Issues:** Since I worked as a single person in the RS section under the Development branch, the conflicts that occurred because RS has a modular structure did not harm the Black Widow side. The way I followed at this point was to merge the project when starting development and then push my own developments. During the development process, I requested a total of 8 merge requests.

○ **Additional Information:**

Lines of Code	1361
Closed Issue Count	44
Open Issue Count	0
Commit Count	63

References of Contribution:

Contributions

All Commits

Closed Issues

Closed Issues for Milestone 2

Individual Contribution:

○ **Member:** Mücahit Uğur

○ **Responsibilities:** Frontend Development

○ **Main contributions:** My primary contributions to the team revolve around the development and enhancement of the application's frontend. This includes not only building and refining the user interface but also rigorously identifying and rectifying any frontend bugs that arise. My role extends to the critical review of pull requests, which encompasses assessing and approving various updates. These updates can range from major deployments to minor bug fixes and other essential improvements. This comprehensive involvement ensures that the frontend operates seamlessly, enhancing user experience and maintaining the application's overall quality

○ **Code-related significant issues:** Since I am working at Frontend development team, I was solving frontend issues. These issues are; Implantation time resolution when adding story, Invalid time value error, don't show null values in story detail,

○ **Management-related significant issues:** I am creating frontend related bugs. Also I am creating meeting notes.Creating user scenario videos.

○ **Pull requests:** As a member of the Frontend Development Team, my responsibilities include creating and reviewing frontend pull requests. When I initiate a pull request, I typically provide a demo of the branch to the reviewers for testing. Once they confirm everything functions correctly, I proceed with the merge. Conversely, when I'm assigned to review a pull request, my focus is on thoroughly testing the featured updates included in it.

References of Contribution:

Contributors

Closed Issues

Created Issues

Pull requests

Lines of Code	316
Closed Issue Count	19
Open Issue Count	1
Pull Request	7
Commit Count	6

Individual Contribution:

- **Member:** Mustafa Görkem Kuyucu
- **Responsibilities:** Web Frontend Development
- **Main contributions:** Even though I have no experience on react web frontend, I started to develop since it is needed in the team. My main contributions are to the react frontend. I developed some of the main features for M2 delivery. I tracked the issues and labeled some of the duplicates. I warned some of my teammates if the issue they opened was already existed. I also reviewed/tested some of the pull requests.
- **Code-related significant issues:** My significant issues (please refer to the link below), which can be observed from the link below, contains edit story component addition (currently it is showing an error with the newly implemented components), tag component enhancements, tag search on the frontend side, using user's location feature, recommendation and timeline page developments, new message component integration, navigation bar is enhancement, QR redirecting component, bug fixes about null values, enhancements about UI forms. Other than that I also included Ant Design and used some of its components.
- **Management-related contributions:** I actively joined lessons and meetings, gave my opinions on the topics and tried to find the best way to proceed. I took notes during some of the meetings. I helped some of my teammates on some issues and raised a flag when I understood I could not finish it on time. I tried to clarify the issues from getting feedback from its creator.
- **Pull requests:** For my pull requests, please refer to the link below. I tried to give as much as the description on the pull request. Some of my pull requests contain more than one commit but they are explained both in the related issue and pull request. I responded quickly to issues I mentioned and clarified the subject. I experienced one conflicted issue which was caused because of the branching from the main branch. Since it was too confusing and nobody could possibly fix it, I closed this pull request and wrote the whole code again to submit it with another pull request.
- **Additional information:** I tried to find manpower to frontend web team since developments were moving slowly and there were not enough people working on it. At the end, we managed to complete most of the deliverables as a team. I also took notes of customer feedback to plan final deliverables.

References of Total Contribution:

Contribution after M1

Closed Issues after M1

Created Issues after M1

Pull requests after M1

Lines of Code for M2	2,441 ++ 158 –
Closed Issue Count	30 (22 after M1)
Open Issue Count	16 (16 after M1)
Pull Request	9 (9 after M1)
Commit Count	26 (25 after M1)

Individual Contribution:

○ **Member:** Gökalp Ayaz

○ **Responsibilities:** Cloud development (web apps, servers, db) , automation (CI/CD, sonarcloud), backend development, frontend development

○ **Main Contributions:** Initially I mostly worked on automation and cloud development. For cloud i deployed 4 web applications (2 web apps for test and 2 web apps for prod environments). I created a server where we host 2 databases (one for test and one for prod). I have set up sonarqube to make sure we have high quality code without security issues. I supported the team to solve CORS and CSRF issues. After the first milestone more issues were present in the Frontend web application. Therefore even though it is not one of my main responsibilities, I focused on frontend issues. I also act as a beta tester for mobile applications. I thoroughly test the application so that no stones are left unturned.

○ **Code Related Issues:** Between first and second milestone, my main focus was frontend development. Here I have modernized some pages with usage of bootstrap components. I have added support for map tools (circle, polyline, polygon). I have improved the way locations are shown in the map as well as on the lists. I have made react updates to improve the way locations are handled wherever a map is shown. I have created new components to improve code reuse. I have added client side validations. I made many little changes such as fixing notification positions, overlapping components, resized images. Please note that I am not listing what I have done for milestone 1 here.

○ **Management Related Issues:** I have reviewed the work of my peers and tried to assist them with their issues and shared possible improvements. I tried to review as many pull requests as possible.

○ **Additional Information:**

Lines of Code	1711++ 1366--
Closed Issue Count	30
Open Issue Count	1

Commit Count	71

References of Contribution:

Contributions

All Commits

Closed Issues

Closed Issues for Milestone 2

The Software:

For the mobile app installation, the user simply has to download the .apk file and install it. After that as a new user, you could register and login and try to add a story. You can find the .apk folder under 0.2.0-alpha pre-release on our github repository. Link: https://github.com/enshkn/BOUN-SWE-574-Fall-23-G2/releases

The web application is up and running on Dutluk App - gcp

Our project contains 5 images that are created via Dockerfile for our React,js frontend, nginx for reverse proxy for the frontend to work on port 80, Uvicorn FastAPI rec engine(Karadut), and Spring Boot backend and finally the default postgres image pulled from the Docker Hub.

We are able to do this via docker-compose.yml on the dutluk/ directory. The instructions for deploying the project are as follows:

-For the .env file please fill all the fields that are intentionally left blank. Please note that Imgur, Pinecone and Google Maps registration is necessary for api key acquisition.

POSTGRES_USER=(db username)

POSTGRES_PASSWORD=(db password)

POSTGRES_DB=(db name)

DUTLUK_DB_URL=(db url)

DB_USERNAME=(db username)

DB_PASSWORD=(db password)

REACT_APP_BACKEND_URL=http://localhost:8080 (placeholder)

REACT_APP_FRONTEND_URL=http://localhost:3000 (placeholder)

JWT_SECRET_KEY=(placeholder)

REACT_APP_GOOGLE_MAPS_API_KEY=(placeholder)

TOKEN_EXPIRATION_HOUR=(in how many hours you want the token to expire)

IMGUR_CLIENT_ID=(imgur api key)

PINECONE_API_KEY=(pinecone api key)

ENVIRONMENT=(placeholder)

PROJECT_INDEX=(placeholder)

REC_URL=http://localhost:8000 (placeholder)

REC_ENGINE_STATUS=false (whether rec engine is active or not)

-For the web application, to run the project on an ubuntu instance please refer to these commands one by one. Note that the server has to have at least 20 gb of memory and 16 gbs of RAM.

sudo su -

apt-get update

apt-get install docker-compose

apt-get install git

git clone https://github.com/enshkn/BOUN-SWE-574-Fall-23-G2

git checkout

cd BOUN-SWE-574-Fall-23-G2/dutluk/dutluk_frontend/

nano .env (paste the necessary env variables here)

cd .. (return to the dutluk/ directory)

cd dutluk_rs/

nano .env (also paste the necessary env variables(PINECONE_API_KEY,ENVIRONMENT,PROJECT_INDEX here)

cd .. (return to the dutluk/ directory)

docker compose up --build -d