

MILESTONE

REPORT 3

31.01.2021

PLATON

THE ACADEMIC COLLABORATION PROGRAM

BACKEND | HALİL UMUT ÖZDEMİR, HÜSEYİN CAN
BÖLÜKBAŞ, HİLAL DEMİR, ALPEREN DİVRİKLİOĞLU
ANDROID | BURAK ÖMÜR, ERTUĞRUL BÜLBÜL, ÖYKÜ
YILMAZ, ORKAN AKIŞU

FRONTEND | AHMET DADAK, RAMAZAN YURT,
BURHAN AKKUŞ, UMUTCAN UVUT

GROUP 7

CMPE 451

Table of Contents

FINAL PROJECT ASSESSMENT	3
INTRODUCTION.....	3
WHERE ARE WE.....	3
Basics.....	3
Planning.....	3
LESSONS LEARNED	4
DELIVERABLES	5
LIST OF DELIVERABLES	5
EVALUATION OF DELIVERABLES.....	5
BACKEND.....	5
FRONTEND	7
ANDROID.....	8
WORK DONE.....	9
SUMMARY OF WORK DONE.....	9
BACKEND.....	9
FRONTEND	10
ANDROID.....	11
UNIT TESTS.....	11
HILAL DEMIR (BACKEND)	11
HALIL UMUT OZDEMIR (BACKEND).....	13
HUSEYIN CAN BOLUKBAS (BACKEND).....	17
ALPEREN DIVRIKLIOGLU (BACKEND).....	Error! Bookmark not defined.
AHMET DADAK (FRONTEND)	19
ACTIVITY STREAM IMPLEMENTATION & W3C STANDARD COMPLIANCE	20
Introduction	20
Activities.....	20
1- Quit Workspace	20
2- Become Contributor	20
3- Create Workspace.....	20
4- Delete Workspace.....	20
5- Create Issue.....	20
6- Create Milestone	20

7- Follow.....	21
8- Upload File	21
9- Create Folder	21
10- Delete File	21
11- Delete Folder	21
12- Comment & Rate User	21
Implementation	22
W3C Activity Stream Protocol Compliance.....	25
API DOCUMENTATION	26
REQUIREMENTS	85
DESIGN DOCUMENTS.....	92
CLASS DIAGRAM.....	92
USE CASE DIAGRAM.....	93
SEQUENCE DIAGRAMS.....	94
MANUALS.....	105
USER MANUAL	105
WEB.....	105
ANDROID.....	155
SYSTEM MANUAL.....	193
BACKEND.....	193
FRONTEND	195
ANDROID.....	196
PROJECT PLAN.....	197
LESSONS LEARNED FROM CUSTOMER MEETING	201

FINAL PROJECT ASSESSMENT

INTRODUCTION

Platon is an academic collaboration platform open to anyone who has an interest in joining an academic research project. It allows its users to create their own profiles, start academic research projects and keep track of them, find suitable collaborators for their projects and follow other researchers they're interested in. It is available on the web and Android, and it is free to use.

WHERE ARE WE

Basics

We've reached the final milestone and we are proud to say that we have successfully implemented every planned feature on our project plan.

Here's a summary of the currently existing features in our platform:

Our users can sign up to our platform and after verifying their accounts through the verification email we send, they can use our platform. They can add personal information such as their names and surnames, their jobs, the institutions they work, their Google Scholar and ResearchGate profiles (the research of a user on those websites gets automatically uploaded to our platform), the skills they have to their profiles. They can set their profiles as "private" or "public". Private profiles can be found via user search; but they're fully visible only to their followers. Public profiles can be seen by both users and guests. A user can delete their account when they want.

Users can follow each other, invite each other to their workspaces to collaborate, rate and comment on each other if they have worked before.

Users can search for other users, workspaces and upcoming events on our platform.

If a user wants to start an academic research project, they can create a workspace. Other users can apply to collaborate in that workspace. Collaborators of a workspace upload files to that workspace, create issues to keep track of their tasks.

Our system provides a collaborator recommendation system for the users who are looking for collaborators, a workspace recommendation system for the users who are looking for workspaces to participate, a profile recommendation system for the users to find similar people to follow, an activity stream to allow each user see the updates of the people and the workspaces they follow, a personal reminder system where each user can see the approaching deadlines of their responsibilities.

Planning

By the time we reached the second milestone, we had actually finished the implementations of the big majority of our features, and also that our customer appreciated our work we presented on the second milestone, we did not have to rush before the third milestone. So, we confidently and calmly completed the remaining planned features for the third milestone.

LESSONS LEARNED

During the eight months we have worked together, we have learned several lessons about software development, team and project management:

- Proper communication is vital. Working as a team differs a lot from working as an individual in the ways that it requires to follow each other's work, the decisions of the team and the requests of the customer. So, a good communication plan is always necessary.
- Friendliness within the team is the key. Delivering a product can be a stressful job, especially when there are strict deadlines, so, it is easy for everyone to get stressed and anxious. During the times of stress, helping and supporting each other, involving in each other's work, taking more responsibility become very important to strengthen the team spirit and successfully deliver the product.
- Being careful with time is necessary. As most of us have never experienced to work in such a project where we had to create everything from scratch on our own before, we rarely could estimate the exact time needed for each task. So, not to find ourselves rushing and possibly making mistakes and leaving other team members who are dependent on us worrying, it is important to start early and not take risks of missing the deadline or delivering a partially implemented product.
- Git is critical. Even though we tried to delegate the tasks in a way to prevent conflict of areas, we still had several times when the works of different team members conflicted and without the help of Git and GitHub, they would be very difficult to manage. So, to update a code base properly and keep track of each change, Git is very important in team projects. (in individual projects too!)
- Code review is important. Having our code checked by another eye can increase its readability and efficiency a lot (and it did in our case!), and therefore makes a positive contribution to the whole team. So, even though people might not enjoy it a lot, teammates should always review each other's code and give honest feedback for their own benefit.
- Choosing good tools and libraries matters. Depending on the libraries and tools developers might choose to utilize, they might experience a variety of feelings from "very enjoyable" to "distressing" during the software development process. For example, using Visual Studio Code reduced the possibly longer times we could have devoted for bug fixing with the debugger it provides.

DELIVERABLES

LIST OF DELIVERABLES

	DELIVERABLE	DUE DATE	STATUS
1	PROJECT PLAN		DELIVERED
2	BACKEND		
	Recommendation System	26.01.2021	DELIVERED
	Tag Search System	26.01.2021	DELIVERED
	Report System	26.01.2021	DELIVERED
	Activity Stream	26.01.2021	DELIVERED
	Personal Reminder	26.01.2021	DELIVERED
	E-Mail and in-app Notification System	26.01.2021	DELIVERED
3	FRONTEND		
	Recommendation System	26.01.2021	DELIVERED
	Tag Search System	26.01.2021	DELIVERED
	Report System	26.01.2021	DELIVERED
	Activity Stream	26.01.2021	DELIVERED
	Personal Reminder	26.01.2021	DELIVERED
	E-Mail and in-app Notification System	26.01.2021	DELIVERED
4	ANDROID		
	Recommendation System	26.01.2021	DELIVERED
	Tag Search System	26.01.2021	DELIVERED
	Report System	26.01.2021	DELIVERED
	Activity Stream	26.01.2021	DELIVERED
	Personal Reminder	26.01.2021	DELIVERED
	E-Mail and in-app Notification System	26.01.2021	DELIVERED

EVALUATION OF DELIVERABLES

BACKEND

Recommendation System

Explanation

There are 3 types of recommendation in our system which are follow recommendations, collaboration recommendations, and workspace recommendations. All three types of recommendations were implemented.

Covered Requirements

1.2.1.3, 1.2.1.4, 1.2.3.1, 1.2.3.2, 1.2.3.3, 1.2.3.4

Tag Search System

Explanation

A system which enables users to search all users or workspaces in our system according to tags that they have. This has no requirements, we optionally added this functionality.

Report System

Explanation

We implemented a report system which enables users to report inappropriate behaviors of other users that they collaborated before. After the report process the information of the report and suspend and ban options are send to admins by using their e-mails.

Covered Requirements

1.1.3.5

Activity Stream

Explanation

An activity stream which shows most of in-app activities of the users that a user follow or collaborated with is implemented. The activity types that appears about the users that a user collaborated with are new collaborators of their workspaces, issue creation, file and folder creation, milestone creation, file and folder deletion. For the following users case, activity types are quitting from a workspace, creation and deletion of a workspace, starting to follow another user, creating a comment.

Covered Requirements

1.2.1.1, 2.5.5

Personal Reminder

Explanation

A personal reminder which includes the deadlines of the workspaces, issues and milestones that are related a user is implemented. This has no requirements, we optionally added this functionality.

E-Mail and in-app Notification System

Explanation

There were some missing notifications at the Milestone 2, these notifications are added to the system. Also a HTML templates are added to the all e-mails including notification, account verification and reset password. Also, the ability to turn off notification was implemented.

Covered Requirements

1.1.5.6, 1.2.4.1, 1.2.4.3, 1.2.4.4, 1.2.4.5

FRONTEND

Recommendation System

Explanation

Users can see the workspace recommendations on his/her workspace page. Users can see the collaborator recommendations on his/her workspace details page. Users can see follow recommendations on his/her profile page.

Covered Requirements

1.2.1.3, 1.2.1.4, 1.2.3.1, 1.2.3.2, 1.2.3.3, 1.2.3.4

Tag Search System

Explanation

Users can search other users and workspaces using tags on the search page. Users can search a workspace with tags on the workspace details page. This has no requirements, we optionally added this functionality.

Report System

Explanation

Users can report other users that misbehave in a collaboration.

Covered Requirements

1.1.3.5

Activity Stream

Explanation

Users can see the activity stream which shows most of the in-app activities of the users that a user follows or collaborates with on the home page.

Covered Requirements

1.2.1.1, 2.5.5

Personal Reminder

Explanation

Users can see a personal reminder which includes the deadlines of the workspaces, issues and milestones in the homepage, and the workspace page. This has no requirements, we optionally added this functionality.

E-Mail and in-app Notification System

Explanation

Users can be notified of the activities in the app via e-mail and in-app notification system. Users can turn off these notifications.

Covered Requirements

1.1.5.6, 1.2.4.1, 1.2.4.3, 1.2.4.4, 1.2.4.5

ANDROID

Recommendation System

Explanation

There are 3 types of recommendation in our system which are follow recommendations, collaboration recommendations, and workspace recommendations. You can see the follow recommendations in the followers or following section, workspace recommendations on the list of your workspaces page, user recommendations on the page of the workspace. This is fully delivered.

Covered Requirements

1.2.1.3, 1.2.1.4, 1.2.3.1, 1.2.3.2, 1.2.3.3, 1.2.3.4

Tag Search System

Explanation

A system which enables users to search all users or workspaces in our system according to tags that they have. This has no requirements, we optionally added this functionality. If you touch the skill on a user's profile, you search for users, if you touch on a skill on a workspace, you search among the workspaces.

Report System

Explanation

We implemented a report system which enables users to report inappropriate behaviors of other users that they collaborated before. After the report process the information of the report and suspend and ban options are send to admins by using their e-mails. To report, yo need to go to the profile and expand the comment section.

Covered Requirements

1.1.3.5

Activity Stream

Explanation

An activity stream which shows most of in-app activities of the users that a user follows or collaborated with is implemented. It is on the home page of the user.

Covered Requirements

1.2.1.1, 2.5.5

Personal Reminder

Explanation

A personal reminder which includes the deadlines of the workspaces, issues and milestones that are related a user can be viewed by clicking the calendar icon on the right top of the home page. This has no requirements, we optionally added this functionality.

E-Mail and in-app Notification System

Explanation

To mute notifications, user must go to their profile and touch on the regarding button. This is fully delivered.

Covered Requirements

1.1.5.6, 1.2.4.1, 1.2.4.3, 1.2.4.4, 1.2.4.5

WORK DONE

SUMMARY OF WORK DONE

BACKEND

TEAM MEMBER	CONTRIBUTION
<i>HALİL UMUT ÖZDEMİR</i>	Tag Search is implemented. Recommendation System and its unit tests are implemented. HTML templates are added to e-mails Notifications are completed and e-mail notification added. Muting Notification is added. Bug fixes and code reviews.
<i>HÜSEYİN CAN BÖLÜKBAS</i>	Implemented Activity Stream API(GET) and its unit tests. Modified existing APIs of File System, Workspace System and Follow System to be able to create Activity System Records. Kept track of the Github issues and reviewed pull requests of the backend team.
<i>HİLAL DEMİR</i>	Implemented the Report System APIs (GET, POST, DELETE) and its unit tests. Kept track of the Github issues and reviewed pull requests of the backend team.
<i>ALPEREN DİVRİKLİOĞLU</i>	Implemented the API for personal reminder of each user.

FRONTEND

TEAM MEMBER

TEAM MEMBER	CONTRIBUTION
<i>HASAN RAMAZAN YURT</i>	Implemented user report interface Implemented tag search interface Implemented user recommendation section in profile page Implemented permission section in user edit page Fixed bugs related with search and report interfaces
<i>AHMET DADAK</i>	Implemented issue section Implemented comment section in issues Implemented workspace recommendation section Implemented collaborator recommendation section Implemented image preview in file section Implemented workspace search with tags in workspace view section Implemented tests for the components Styled UI and fixed bugs Deployed frontend
<i>BURHAN AKKUS</i>	Implemented Activity Stream Added notification activation/deactivation to profile settings.
<i>UMUTCAN UVUT</i>	Implemented personal reminder section (calendar widget substitute at the project plan). Implemented comment/rate section. Fixed a bug on the comment section at workspace issues. Fixed a bug related to pagination use at upcoming events and trending projects. Various UI improvements.

ANDROID

TEAM MEMBER	CONTRIBUTION
BURAK ÖMÜR	Implemented Tag search for Users and Workspaces Implemented the reporting user option Contributed to the implementation of Personal Calendar
ERTUĞRUL BÜLBÜL	Implemented delete issue comment Lottie animation changed Implemented Activity Stream
ÖYKÜ YILMAZ	Implemented recommendation for following users, workspace recommendation for users, user recommendations for workspaces. Implemented the personal calendar with Burak. Implemented mute notifications, both in-app and email notifications.
ORKAN AKİŞÜ	

UNIT TESTS

HILAL DEMİR (BACKEND)

Report System Unit Tests (sha: 8b5b708c8062bbbc769eb08ca64bd7030e610a3c and b1cbe82d38a34cdaa5ac924f136ff9febb66a952):

I implemented 4 unit tests for the Report System. All tests have passed with their expected results and expected status code which is 200.

1. **(POST) test_post_report_valid:** It is for posting a valid report to a user. Test generates a valid token for user 1, creates the data for reporting user 2 and expected response which is 'msg': 'Report addition is successfully sent via email'. Then it gets the actual response with data and authentication token which are required by the POST method of the Report System. Test passes as it should be because user 1 has collaborated with user 2 and has not reported him before. So he can report him.
2. **(POST) test_post_report_invalid:** It is for posting an invalid report to a user. Test generates a valid token for user 1, creates the data for reporting user 3 and expected response which is 'error': 'You can not send comments to the user which you did not worked before'. Then it gets the actual response with data and authentication token which are required by the POST method of the Report System. Test passes as it should be because user 1 has not collaborated with user 3 so he cannot report him.
3. **(GET) test_get_reports:** It is for getting reports made by a user. Test generates a valid token for user 1 and sends the request for getting the reports. The test passes with status code 200 because the GET method of the Report System only requires a valid token and gets the information about the user from this token.
4. **(DELETE) test_delete_report:** It is for deleting reports made by a user. Test generates a valid token for user 1 and sends the request for deleting the report with the report id. The test passes with status code 200 because user 1 has posted a report with the id 1 before. Therefore the system was able to delete it with the valid token.

Upcoming Event Search Unit Tests (sha: 40a7007f3da8447fb211035e3a12375284df9d6c):

I implemented 4 unit tests for Upcoming Events. All tests have passed with their expected results and expected status codes which is 200.

1. **(GET) test_name_search** for searching a keyword in the names of the upcoming events. Keyword “simulation” is passed as a searching query and the expected result is the two of the upcoming events contain “simulation” in their event names from the test environment database. The test has passed.
2. **(GET) test_location_search** for searching a keyword in the location of the upcoming events. Keyword “malta” is passed as a search query. It may seem like the result should contain only one event but since there is semantic search in the system, “portugal” keyword is also searched in the upcoming events. So, the expected result is two of the upcoming events contain “malta” or “portugal” in their location names from the test environment database. The test has passed.
3. **(GET) test_deadline_search** for advanced search with specific deadlines. Keyword “simulation” is passed as a searching query and (2020,12,1,23,59,59) datetime data is passed as deadline_filter_start. This searches the events that contain simulation in their names and filters the results which their deadlines start after the deadline_filter_start. There are 2 results that pass these conditions. Expected result is true so the test has passed.
4. **(GET) test_sort** for sorting the results. Keyword “simulation” is passed as a searching query and “1” is passed as sorting_criteria which means sorting according to starting dates. In the test_name_search test the results were displayed according to semantic rating. In this test, results are same but their appearance is different since this time results are sorted according to their starting dates. Expected result is true so the test has passed.

User Skill Unit Tests (sha: c6e10808b68dd8c54648a1ad256fa82ce5fe537a):

I implemented 1 unit test for User Skills. The test has passed with its expected result and expected status code which is 200.

1. **(POST) test_post_userskill** for posting a skill into a user’s profile. The test generates a valid token since the POST endpoint requires authentication token as header. Then it sends the skill “javascript” with the valid token to the endpoint and receives the response which is ‘Skill is successfully added’. Expected result is true so the test has passed.

HALIL U MUT OZDEMIR (BACKEND)

Base Test(sha: Off7b14bd97b45db572debd408979164fe464887)

I implemented a Base test class to configure the test database and a testing mechanism which removes all of the contents of the test database after each test case.

Login Unit Tests (sha: 83d5c59890936211a0e67c4ba267db14da0e80b4, fbd9fa97e94fa36da9380994d778c0a16c39fb01)

I implemented 3 test cases to validate the functionality of the login endpoint. At the beginning of each test I created artificial data on the test database. Artificial data contains possible users of the system which contains different properties with respect to account validity.

1. `test_valid_login`: Checks the login endpoint with the valid data. The expected result is a valid token and 200 HTTP status code.

2. `test_invalid_login`: Checks the login endpoint with the invalid data. Given password is not the real password of the user. The expected result is an error message and 401 HTTP status code.

3. `test_not_activated_user`: Checks the login endpoint with the valid data. But the user which has

these credentials are not valid. Therefore, the expected result is an error message and 401 HTTP status code.

Reset Password Test(sha: 2005e36d68770455103853963efe8ab49ff6e6e4)

I implemented 4 test cases to validate the functionality of reset password endpoint. At the beginning of each test I created artificial data on the test database. Artificial data contains possible users of the system which contains different properties with respect to account validity.

1. `test_reset_password_valid`: Checks the reset password endpoint with the data. The expected result is a valid response message and 200 HTTP status code.

2. `test_reset_password_invalid_input`: Checks the reset password endpoint with the invalid data.

In the given data, the passwords do not match. The expected result is an error message and 400 HTTP status code.

3. `test_reset_password_invalid_input2`: Checks the reset password endpoint with the invalid data.

In the given data password is not written twice. The expected result is an error message and 400 HTTP status code.

4. test_reset_password_invalid_token: Checks the reset password endpoint with the invalid data.

In the given data, the given token is not valid. So it is an unauthorized call. The expected result is an error message and 401 HTTP status code.

Research Information Tests(sha:

8fedb92915d181a3ccfcf437612869e913b43b7, 81ef390fa900a24f5af2f9982f9f3033844b300c)

I implemented 6 test cases to validate the functionality of the research information system. At the beginning of each test I created artificial data on the test database. Artificial data contains possible users of the system and their possible research information in our system.

1. test_add_research_info_valid: Checks the POST request of research information endpoint with

the valid data. The expected result is a valid response message and 201 HTTP status code. Also new record must be added to the test database.

2. test_add_research_info_invalid: Checks the POST request of research information endpoint with

the invalid data. The year information is missing in the data. The expected result is an error message and 400 HTTP status code. Also new record must not be added to the test database.

3. test_remove_research_info_valid: Checks the DELETE request of research information endpoint

with the valid data. The expected result is a valid response message and 201 HTTP status code. Also new record must be deleted from the test database.

4. test_remove_research_info_invalid: Checks the DELETE request of research information endpoint with the invalid data. The owner of the token and the owner of the research information is not the same. The expected result is an error message and 400 HTTP status code. Also new record must not be deleted from the test database.

5. test_fetch_RG_info: Checks the fetch ResearchGate information with artificial data. The expected result of the function is specified in the unit test.

6. test_fetch_GS_info: Checks the fetch GoogleScholar information with artificial data. The expected result of the function is specified in the unit test.

Notification Tests (sha: 47c82110470d5fc81ab29999da57621aefbea4de)

I implemented 5 test cases to validate the functionality of the notification system. At the beginning of

each test I created artificial data on the test database. Artificial data contains possible users of the system and their possible notifications in our system.

1. `test_add_notification`: Checks the POST request of notification endpoint with the valid data. The

expected result is a valid response message and 201 HTTP status code. Also new record must be added to the test database

2. `test_get_notification_valid`: Checks the GET request of notification endpoint with the valid data.

The expected result is the list of notifications and 200 HTTP status code.

3. `test_get_notification_invalid`: Checks the GET request of notification endpoint with the invalid data. Given token does not have the true format. The expected result is an error message and 401 HTTP status code.

4. `test_delete_notification_valid`: Checks the DELETE request of notification endpoint with the valid data. The expected result is a valid response message and 200 HTTP status code. Also new record must be deleted from the test database.

5. `test_delete_notification_invalid`: Checks the DELETE request of notification endpoint with the invalid data. Given notification id does not exist in the test database. The expected result is an error message and 404 HTTP status code. Also new record must not be deleted from the test database.

Search History Tests (sha: 434401ae41de337d11834db24d4f12715fcf1e5b)

I implemented 4 test cases to validate the functionality of the search history system. At the beginning of each test I created artificial data on the test database. Artificial data contains possible users of the system and their possible search history in our system.

1. `test_user_search_history`, `test_workspace_search`, `test_ue_search_history`: Checks the GET request of search history endpoint for 3 types of search in our system with the valid data. The expected result is the list of search history ordered by number of use and 200 HTTP status code.

2. `test_empty_result`: Checks the GET request of search history endpoint for the empty response.

The expected result is an empty list and 200 HTTP status code.

User Search Tests (sha: 7bf8dd85144c6edb8fd4cef8453e70435d9c1a10)

I implemented 7 test cases to validate the functionality of the search engine. At the beginning of each test I created artificial data on the test database. Artificial data contains possible users of the system and their possible search history in our system.

1. `test_name_search`: Checks the search engine by searching a user with the name of this user.
2. `test_no_result_search`: Checks the search engine by a search query which does not have any related user.
3. `test_no_result_search_filter`: Checks the search engine by a search query which does not have any related user. The reason why there is no related user is the given filters to the search engine.
4. `test_skill_search`: Checks the search engine by searching a user with s skill of this user.
5. `test_job_search`: Checks the search engine by searching a user with the job of this user.
6. `test_semantic_search`: Checks the search engine by searching a user with the "Schoolman" search query. Because a user has a job of "PhD Student" this user matches with the search query.
7. `test_institution_search`: Checks the search engine by searching a user with the institution of this user.

Upcoming Events Test (sha: 67e3a56568a99e746f6fae4ad0553fefec0f1ac)

I implemented 3 test cases to validate the functionality of the upcoming events system.

1. `test_parse_event`: Checks the function which parses an upcoming event from the HTML of the given event.
2. `test_get_number_of_pages`: Checks the function that fetches the number of pages of the website that upcoming events are fetched.
3. `test_parse_page`: Checks the function that fetches the upcoming events with the given page index.

Recommendation System Test (sha: fd3a2cb434b7ce5e355dfe025ea3badd8636e1b)

I implemented 3 test cases to validate the functionality of the 3 types of recommendations that Recommendation System mst give recommendations.

1. `test_workspace_recommendations`: Checks the quality of workspace recommendations and also the bugs of the function that calculates recommendations periodically.

2. test_follow_recommendations: Checks the quality of follow recommendations and also the bugs of the function that calculates recommendations periodically.

3. test_collaboration_recommendations: Checks the quality of collaboration recommendations and also the bugs of the function that calculates recommendations periodically.

HUSEYIN CAN BOLUKBAS (BACKEND)

Activity Stream Unit Tests (sha: 3c65011780b2c8fc12576ab6a5ce8ae8f3b49aa4)

Unit tests checks the following activities:

- **Follow:** Whenever a user follows another user either directly or after his/her follow request is accepted, an activity record is created and shown to the users that follow him/her.
- **Rate & Comment:** Whenever a user comments and rates another user, an activity record is created and shown to the users that follow him/her.
- **Create Workspace:** Whenever a user creates a workspace, an activity record is created and shown to the users that follow him/her.
- **Leave Workspace:** Whenever a user quits the workspace that he/she contributes to, an activity record is created and shown to the users that follows him/her.
- **New Contribution:** Whenever a user accepts a workspace invitation or that user's workspace application is accepted, an activity record is created and shown to the users that are contributing to that workspace.
- **Delete Workspace:** Whenever a user deletes a workspace that he/she actively contributes to, an activity record is created and shown to the users that follow him/her.

Workspace Search (sha: 516b83e5065472bda6fa0f26d547b8b5a1807b02)

Unit tests checks below cases:

- if the searched element appears in the workspace title.
 - if searched element is one word query
 - if searched element is multiple words
 - if searched element is multiple words and contains white spaces
- if searched element appears in the workspace description
 - if searched element is one word query
 - if searched element is multiple words
 - Check if upper case letters are ignored while searching.
- Check if private workspaces do not appear in the search.
- Check if the searched element does not appear in the title but is similar to a word in title semantically.
- Check if the searched element does not appear in description but is similar to a word in description semantically.
- Check if the searched element does not appear in the title or description. It should return no values.
- Check if the skill filter works.
 - Check that without a skill, workspace_ids 3 and 5 is found with "bravest" query.
 - Check that with a skill, only workspace_id=5 is found with "bravest" query.
- Check if the starting date filter works.
 - Check that without a date filter, workspace_ids 3 and 5 is found with "bravest" query.
 - Check that with a date filter, nothing is found.

- Check if the deadline filter works.
 - Check that without a deadline filter, workspace_ids 3 and 5 is found with "bravest" query.
 - Check that with a deadline filter, only workspace_id=5 is found with "bravest" query.
- Check if sorting criterias work.
 - check if sorting with respect to date works
 - check if sorting with “number of collaborators needed” works
 - check if sorting with alphabetical order works
- Check if the founder name and surname filter works.

Milestone API (sha: 0a43f2a32a95e08ecdc2bf66afa662fd58f26e83)

Unit tests checks below cases:

- Can will try to get the milestones of a public workspace in which there is no milestone.
- Can will try to get the milestones of a public workspace. It should be successful.
- Umut will try to get the milestones of a private workspace. It should be unsuccessful.
- Can will try to get the milestones of a private workspace in which he is an active contributor. It should be successful.
- Can will try to get the milestones of a private workspace in which he is not an active contributor anymore. It should be unsuccessful.
- Can will try to post a milestone into a workspace in which he is a contributor. Success.
- Can will try to post a milestone into a workspace in which he is not a contributor. Fail.
- Can will try to update a milestone at a workspace in which he is a contributor. Success.
- Can will try to update a milestone at a workspace in which he is not a contributor. Fail.
- Can will try to delete a milestone at a workspace in which he is a contributor. Success.
- Can will try to delete a milestone at a workspace in which he is not a contributor. Fail.

Issue API (sha: 3d089c8069ef6027ce743e6c3db2f8ba1ec4ffbb)

Unit tests checks below cases:

- Can will try to get the issues of a public workspace. It should be successful.
- Umut will try to get the issues of a private workspace. It should be unsuccessful.
- Can will try to get the issues of a private workspace in which he is an active contributor. It should be successful.
- Can will try to get the issues of a private workspace in which he is not an active contributor anymore. It should be unsuccessful.
- Can will try to post an issue into a workspace in which he is a contributor. Success.
- Can will try to post an issue into a workspace in which he is not a contributor. Fail.
- Can will try to update an issue at an workspace in which he is a contributor. Success.
- Can will try to update an issue at a workspace in which he is not a contributor. Fail.
- Can will try to delete an issue at a workspace in which he is a contributor. Success.
- Can will try to delete an issue at a workspace in which he is not a contributor. Fail.

Issue Assignee API (sha: 3d089c8069ef6027ce743e6c3db2f8ba1ec4ffbb)

Unit tests checks below cases:

- Can will try to get the issue assignees at a public workspace. Success.
- Can will try to get the issue assignees at a private workspace. Fail.

- Can will try to get the issue assignees at a private workspace in which he is a contributor. Success.
- Hilal will try to post an issue assignee at a workspace in which she is a contributor. Assignee(Umut) is also the contributor of the workspace. Success.
- Can will try to post an issue assignee at an workspace in which he is a contributor. Assignee is not a contributor of the workspace. Fail.
- Can will try to post issue assignee at a workspace in which he is not a contributor. Fail.
- Can will try to post an issue assignee at an workspace in which he is a contributor. However, that person already exists in the issue assignees. Fail.
- Can will try to remove issue assignee at a workspace in which he is a contributor. Success.
- Can will try to remove issue assignee at a workspace in which he is not a contributor. Fail.

Issue Comment API (sha: 3d089c8069ef6027ce743e6c3db2f8ba1ec4ffbb)

Unit tests assures below cases with their expected result:

- Can will try to get the issue comments at a public workspace. Success.
- Can will try to get the issue comments at a private workspace. Fail.
- Can will try to get the issue comments at a private workspace in which he is a contributor. Success.
- Can will try to post issue comments at a workspace in which he is a contributor. Success.
- Can will try to post issue comments at a workspace in which he is not a contributor. Fail.
- Can will try to remove issue comments at a workspace in which he is a contributor. Also it is his comment. Success.
- Can will try to remove issue comments at a workspace in which he is a contributor. However it is not his comment. Fail.
- Can will try to remove issue comments at a workspace in which he is not a contributor. Fail.

Follow System Unit Tests (sha: 100b3d48934942b35029555ccf0fdf881a42d1a5)

Unit tests checks following cases:

- Check if Umut follows Can, Umut's following list should include Can.
- If Umut and Alperen follow Can. Then Can's follower list should contain their IDs.
- Check if incoming follow requests can be seen by the users.
- If Umut sends a follow request to Hilal, Hilal should see Umut's ID in followRequests list.
- Check if the follow request instance is deleted after that follow request is accepted/rejected.
- Check if follow requests can be accepted and rejected successfully.
- Check if unfollow functionality works.

AHMET DADAK (FRONTEND)

Recommendation System Tests (sha: e736179d9bdd80df23c7e8cf9bf22b5aba0af67)

Tests that check the following components:

- **CollaboratorRecommendationDialog:** Tests the Collaborator Recommendation component is rendered correctly or not.
- **WorkspaceRecommendation:** Tests the Workspace Recommendation component is rendered correctly or not.

ACTIVITY STREAM IMPLEMENTATION & W3C STANDARD COMPLIANCE

Introduction

We've built an activity stream system that supports W3C Activity Stream Protocol. Users can see the activities that are related to them as a stream. Properties of the Activity Stream will be explained below. To give a brief overview, our activity stream contains activities of the workspaces that the user contributes to and the activities of the other users that one follows.

Activities

Platon's Activity Stream supports following activities:

1- Quit Workspace

Whenever a user quits the workspace that he/she contributes to, an activity record is created and shown to the users that follows him/her.

e.g. "Ahmet Dadak left Discovering Bias in Datasets workspace"

2- Become Contributor

Whenever a user accepts a workspace invitation or that user's workspace application is accepted, an activity record is created and shown to the users that are contributing to that workspace.

e.g. "Halil Umut Özdemir is now a contributor in Automated Conduction for Symphony Orchestras workspace"

3- Create Workspace

Whenever a user creates a workspace, an activity record is created and shown to the users that follow him/her.

e.g. "Ertugrul Bulbul created a workspace named Spacing in Basketball"

4- Delete Workspace

Whenever a user deletes a workspace that he/she actively contributes to, an activity record is created and shown to the users that follow him/her.

e.g. "Can Bolukbas deleted the workspace Iterated Prisoners Dilemma"

5- Create Issue

Whenever a user creates an issue in the workspace that he/she actively contributes to, an activity record is created and shown to the users that are contributing to that workspace.

e.g. "Umutcan Uvut created an issue titled 'Collect and Annotate Training Data' in the workspace 'Discovering Bias in Datasets'"

6- Create Milestone

Whenever a user creates a milestone in the workspace that he/she actively contributes to, an activity record is created and shown to the users that are contributing to that workspace.

e.g. "Christophe Daussy created a milestone titled 'Train a Neural Network' in the workspace 'Predicting Earthquake with DL'"

7- Follow

Whenever a user follows another user either directly or after his/her follow request is accepted, an activity record is created and shown to the users that follow him/her.

e.g. "Burhan Can Akkus started following Gwen Stacy"

8- Upload File

Whenever a user uploads a file to the workspace, an activity record is created and shown to the users that are contributing to that workspace.

e.g. "Ahmet Dadak uploaded a new file named 'HelperClasses.py' to the workspace 'Unique ID management'"

9- Create Folder

Whenever a user creates a folder in the workspace, an activity record is created and shown to the users that are contributing to that workspace.

e.g. "Gwen Stacy created a new folder named 'Charts' in the workspace 'Predicting Earthquake with DL'"

10- Delete File

Whenever a user deletes a file in the workspace, an activity record is created and shown to the users that are contributing to that workspace.

e.g. "Gwen Stacy deleted the file named 'IMG-20210126-WA0005.jpg' in the workspace 'Predicting Earthquake with DL'"

11- Delete Folder

Whenever a user deletes a folder in the workspace, an activity record is created and shown to the users that are contributing to that workspace.

e.g. "Ahmet Dadak deleted the folder named 'Prediction' in the workspace 'Medical Information System'"

12- Comment & Rate User

Whenever a user comments and rates another user, an activity record is created and shown to the users that follow him/her.

e.g. "Oyku Yilmaz commented and rated Halil Umut Özdemir"

Implementation

Firstly, we've decided which activities that Platon will support. After deciding the activities, a database table is created. Columns of this table can be seen below.

```
1  from app import db
2
3  class ActivityStreamItem(db.Model):
4      __tablename__ = 'activity_stream'
5      activity_id = db.Column(db.Integer,primary_key=True,autoincrement=True)
6      activity_context_vocab = db.Column(db.String(200), nullable=False)
7      activity_context_ext = db.Column(db.String(200), nullable=True)
8      activity_summary = db.Column(db.String(200), nullable=False)
9      activity_type = db.Column(db.String(50), nullable=False)
10     activity_actor_type = db.Column(db.String(50), nullable=False)
11     activity_actor_id = db.Column(db.Integer,db.ForeignKey('users.id',ondelete="CASCADE"), nullable=False)
12     activity_actor_name = db.Column(db.String(50), nullable=False)
13     activity_actor_image_url = db.Column(db.String(100), nullable=False)
14     timestamp = db.Column(db.DateTime,default=db.func.now(),nullable=False)
15     activity_object_type = db.Column(db.String(50), nullable=True)
16     activity_object_id = db.Column(db.Integer, nullable=True)
17     activity_object_name = db.Column(db.String(50), nullable=True)
18     activity_object_image_url = db.Column(db.String(100), nullable=True)
19     activity_object_content = db.Column(db.String(250), nullable=True)
20     activity_object_rating_value = db.Column(db.Integer, nullable=True)
21     activity_target_type = db.Column(db.String(50), nullable=True)
22     activity_target_id = db.Column(db.Integer, nullable=True)
23     activity_target_name = db.Column(db.String(50), nullable=True)
24     activity_target_image_url = db.Column(db.String(100), nullable=True)
```

Note that the above table does not contain all necessary fields of Activity Stream Protocol such as there is no pagination information above.

After creating the database table, existing APIs are modified to be able to create activity stream records. For example, API that handles file upload requests are modified to also create an activity record.

Lastly, Activity Stream API with GET method is created. This method returns the activity stream of a user by following W3C Activity Stream Protocol. Return format of the GET method can be seen in the example below.

```
1 √ {
2   "@context": "https://www.w3.org/ns/activitystreams",
3   "id": 0,
4   "orderedItems": [
5     {
6       "@context": {
7         "@language": "en",
8         "@vocab": "https://www.w3.org/ns/activitystreams",
9         "ext": null
10      },
11      "actor": {
12        "id": 16,
13        "image": {
14          "type": "Image",
15          "url": "/auth_system/profile_photo?user_id=16"
16        },
17        "name": "Christophe Daussy",
18        "type": "Person"
19      },
20      "object": {
21        "content": null,
22        "ext:ratingValue": null,
23        "id": null,
24        "image": {
25          "type": "Image",
26          "url": null
27        },
28        "name": "paper.pdf",
29        "type": "Document"
30      },
31      "summary": "Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'",
32      "target": {
33        "id": 10,
34        "image": {
35          "type": "Image",
36          "url": null
37        },
38        "name": "Predicting Earthquake with DL",
39        "type": "Group"
40      },
41      "type": "Create"
42    },
43  ],
44  "type": "Create"
45 }
```

```

43  {
44    "@context": {
45      "@language": "en",
46      "@vocab": "https://www.w3.org/ns/activitystreams",
47      "ext": "http://schema.org/Rating"
48    },
49    "actor": {
50      "id": 5,
51      "image": {
52        "type": "Image",
53        "url": "/auth_system/profile_photo?user_id=5"
54      },
55      "name": "Oyku Yilmaz",
56      "type": "Person"
57    },
58    "object": {
59      "content": "An average team mate, sometimes works independently and does not inform others.",
60      "ext:ratingValue": 3,
61      "id": 8,
62      "image": {
63        "type": "Image",
64        "url": null
65      },
66      "name": "Comment",
67      "type": "Note"
68    },
69    "summary": "Oyku Yilmaz commented and rated Halil Umut Özdemir",
70    "target": {
71      "id": 13,
72      "image": {
73        "type": "Image",
74        "url": "/auth_system/logo"
75      },
76      "name": "Halil Umut Özdemir",
77      "type": "Person"
78    },
79    "type": "Add"
80  }
81 ],
82 "summary": "Page 0 of Activity Stream",
83 "totalItems": 4,
84 "type": "OrderedCollectionPage"
85 }

```

Please note that

id represents the id of the OrderedCollectionPage.

ext represents the external contexts.

totalItems represents the total number of OrderCollectionPage in the system.

id and **totalItems** fields depend on the API's parameters which are shown below:

Name	Description
page * required integer (query)	Page index that you want(Starts from 0)
per_page * required integer (query)	Number of items in a page
auth_token * required string (header)	Authentication Token

W3C Activity Stream Protocol Compliance

Contexts that are used are

<https://www.w3.org/ns/activitystreams>

<http://schema.org/Rating>

Example activity stream is shown in the “Implementation” section. All fields in the JSON file are defined in their corresponding contexts which is necessary for the protocol. Therefore, we believe that Platon’s Activity Stream complies with the W3C Activity Stream Protocol.

API DOCUMENTATION

Authentication System Authentication System Endpoints

POST /auth_system/login Takes Login Credentials as argument and returns a valid token

Parameters

Name	Description
e_mail * required string (formData)	E-mail of the person
password * required string (formData)	Password of the person

Try it out

Responses

Response content type application/json

Code	Description
200	Valid token
400	Input Format Error
401	Account Problems
404	User not found
500	Database Connection Error

GET /auth_system/logo

Parameters

No parameters

Responses

Response content type application/json

Code	Description
200	Valid Response

Try it out

GET /auth_system/profile_photo

Parameters

Name	Description
user_id * required string (query)	ID of the User

Responses

Code	Description
200	<i>Valid Response</i>
400	<i>Input Format Error</i>
404	<i>Profile Photo is not Found</i>
500	<i>Database Connection Error</i>

Response content type application/json

Try it out

POST /auth_system/reset_password Changes password of the user

Parameters

Name	Description
new_password * required string (formData)	Enter new password
new_password_repeat * required string (formData)	Enter new password(Again)
auth_token * required string (header)	Token that sent via email as a URL link

Responses

Code	Description
200	<i>Password Successfully Changed</i>
400	<i>Passwords are not matched</i>
401	<i>Authorization Error</i>
500	<i>Database Connection/E-mail Server Error</i>

Response content type application/json

Try it out

GET /auth_system/reset_password Sends a reset password email to the user

Parameters

Name	Description
e_mail * required string (query)	E-mail of the person

Responses

Response content type application/json

Code	Description
200	Valid e-mail
	Example Value Model
	{ "msg": "string" }
400	Input Format Error
401	Account Problems
404	E-mail not found
500	Database Connection/E-mail Server Error

GET /auth_system/self

Parameters

Name	Description
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
200	User has been found.
	Example Value Model
	{ "id": 0, "name": "string", "surname": "string", "profile_photo": "string", "e_mail": "string", "job_id": 0 }
404	The user is not found.
500	The server is not connected to the database.

POST /auth_system/skills Adds a new skill to user's skills with name

Parameters

Name	Description
skill * required string (formData)	Skill name
auth_token * required string (header)	Authentication token

Responses

Code	Description
200	Skill is Successfully Added
400	Input Format Error
404	User is not found
500	Database Connection

Response content type application/json

GET /auth_system/skills Returns a list of user's skills with id and name

Parameters

Name	Description
user_id * required integer (query)	ID of the requested user.
auth_token * required string (header)	Authentication token

Responses

Code	Description
200	Skills are Successfully Returned
206	Partial Content
400	Input Format Error
404	Skills are empty
500	Database Connection

Response content type application/json

DELETE /auth_system/skills Deletes the skill from user's skills with name

Parameters Try it out

Name	Description
skill * required string (formData)	Skill name
auth_token * required string (header)	Authentication token

Responses Response content type application/json

Code	Description
200	SKILL is Successfully Deleted
400	Input Format Error
404	SKILL or UserSkill is not Found
500	Database Connection

POST /auth_system/user

Parameters Try it out

Name	Description
e_mail * required string (formData)	E-mail address of the new user
password * required string (formData)	Password of the new user
name * required string (formData)	Name of the new user
surname * required string (formData)	Surname of the new user
job * required string (formData)	Job of the new user
google_scholar_name string (formData)	URL of the Google Scholar page of the user
researchgate_name string (formData)	URL of the ResearchGate page of the user
institution string (formData)	Institution of the user

Responses Response content type application/json

Code	Description
201	User has been successfully created.
206	Partial Content
400	Missing data fields or invalid data.
409	User with the given e-mail address already exists.
500	The server is not connected to the database.
503	The server could not send the account activation e-mail.

GET /auth_system/user Returns the profile information of the requested user

Parameters

Try it out

Name	Description
user_id <small>* required</small> integer (query)	ID of the requested user.
auth_token <small>* required</small> string (header)	Authentication token

Responses

Response content type application/json

Code	Description
200	User has been found.
	Example Value Model
	{ "id": 0, "name": "string", "surname": "string", "profile_photo": "string", "e_mail": "string", "job_id": 0 }
208	Partial Content
400	Missing data fields or invalid data.
404	The user is not found.
500	The server is not connected to the database.

DELETE /auth_system/user

Parameters

Try it out

Name	Description
password <small>* required</small> string (formData)	Password of the user
auth_token <small>* required</small> string (header)	Authentication token

Responses

Response content type application/json

Code	Description
200	User account has been successfully deleted.
208	Partial Content
400	Missing data fields or invalid data.
401	Wrong password.
404	The user is not found.
500	The server is not connected to the database.

PUT /auth_system/user

Parameters

Name	Description
<code>name</code> string (<code>formData</code>)	Name of the user
<code>surname</code> string (<code>formData</code>)	Surname of the user
<code>e_mail</code> string (<code>formData</code>)	E-mail address of the user
<code>job</code> string (<code>formData</code>)	Job of the user
<code>is_valid</code> integer (<code>formData</code>)	The flag that shows whether the user account is activated or not. (0: Inactive, 1: Active)
<code>is_private</code> integer (<code>formData</code>)	The flag that shows whether the user's profile is public or private. (0: Public, 1: Private)
<code>profile_photo</code> file (<code>formData</code>)	Upload File Here
<code>google_scholar_name</code> string (<code>formData</code>)	URL of the Google Scholar page of the user
<code>researchgate_name</code> string (<code>formData</code>)	URL of the ResearchGate page of the user
<code>institution</code> string (<code>formData</code>)	Institution of the user
<code>auth_token</code> * required string (<code>header</code>)	Authentication token

Responses

Code	Description	Response content type
200	<code>Account information has been successfully updated.</code>	application/json
202	<code>Server has received the request but there was no information to be updated.</code>	
206	<code>Partial Content</code>	
400	<code>Missing data fields or invalid data.</code>	
404	<code>The user is not found.</code>	
500	<code>The server is not connected to the database.</code>	

Follow System

Follow System Endpoints

POST /follow/comment Create user comment
[Try it out](#)

Parameters	
Name	Description
commented_user_id * required integer (formData)	ID of the commented user
rate * required integer (formData)	Rate for the User(Between 1 and 5)
text string (formData)	Comment for the User
auth_token * required string (header)	Authentication token

Responses
Response content type application/json

Code	Description
400	<code>Input Format Error</code>
401	<code>Account Problems</code>
404	<code>Not found</code>
500	<code>Database Connection Error</code>

GET /Follow/comment Get Comments
[Try it out](#)

Parameters	
Name	Description
commented_user_id * required integer (query)	ID of the commented user
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token * required string (header)	Authentication token

Responses
Response content type application/json

Code	Description
200	<code>Success</code>
400	<code>Input Format Error</code>
401	<code>Account Problems</code>
404	<code>Not found</code>
500	<code>Database Connection Error</code>

DELETE /follow/comment Deletes an user comment

Parameters

Try it out

Name	Description
comment_id <small>* required integer (query)</small>	ID of the commented user
auth_token <small>* required string (header)</small>	Authentication token

Responses

Response content type application/json

Code	Description
400	<small>Input Format Error</small>
401	<small>Account Problems</small>
404	<small>Not found</small>
500	<small>Database Connection Error</small>

POST /follow/follow_requests Creates FollowRequest record if profile is private, creates Follow record if profile is public

Parameters

Try it out

Name	Description
follower_id <small>* required integer (formData)</small>	ID of the follower. Follower is the one who follows someone.
following_id <small>* required integer (formData)</small>	ID of the following user. Following user is the one who is followed by someone.
auth_token <small>* required string (header)</small>	Authentication Token

Responses

Response content type application/json

Code	Description
200	<small>Follow Request is sent successfully</small>
400	<small>Input Format Error</small>
401	<small>Current user is unauthorized to send Follow Request</small>
500	<small>Database Connection Error</small>

GET /follow/follow_requests Returns a list of dictionaries that contains id, name, surname, e_mail, rate and is_private

Parameters Try it out

Name	Description
following_id * required integer (query)	ID of the following user. Following user is the one who is followed by someone.
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token * required string (header)	Authentication Token

Responses Response content type application/json

Code	Description
200	<i>Follow Requests List is successfully returned</i>
	Example Value Model <pre>{ "number_of_pages": 0, "follow_requests": [{ "id": 0, "name": "string", "surname": "string", "profile_picture": "string", "e_mail": "string", "job_id": 0 }] }</pre>
400	<i>Input Format Error</i>
401	<i>Current user is unauthorized to see the Follow Requests</i>
404	<i>Follow Requests List is empty</i>
500	<i>Database Connection Error</i>

DELETE /follow/follow_requests Takes the follower_id and following_id as inputs and replies the corresponding Follow Request

Parameters Try it out

Name	Description
follower_id * required integer (formData)	ID of the follower. Follower is the one who follows someone.
following_id * required integer (formData)	ID of the following user. Following user is the one who is followed by someone.
state * required integer (formData)	State is 1 if the reply is accept, state is 2 if the reply is reject.
auth_token * required string (header)	Authentication Token

Responses Response content type application/json

Code	Description
200	<i>Follow Request is replied successfully</i>
400	<i>Input Format Error</i>
401	<i>Current user is unauthorized to reply the Follow Request</i>
404	<i>Follow Request not found</i>
500	<i>Database Connection Error</i>

GET /follow/followers Returns a list of dictionaries with id, name, surname, e_mail, rate and is_private informations

Parameters

Name	Description
following_id <small>* required</small> integer (query)	ID of the following user. Following user is the one who is followed by someone.
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token <small>* required</small> string (header)	Authentication Token

Responses

Code	Description
200	Followers List is successfully returned
	Example Value Model
	<pre>{ "number_of_pages": 0, "followers": [{ "id": 0, "name": "string", "surname": "string", "profile_photo": "string", "e_mail": "string", "job_id": 0 }] }</pre>
400	Input Format Error
404	Followers List is empty
500	Database Connection Error

GET /follow/followings Returns a list of dictionaries with id, name, surname, e_mail, rate and is_private informations

Parameters

Name	Description
follower_id <small>* required</small> integer (query)	ID of the follower. Follower is the one who follows someone.
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token <small>* required</small> string (header)	Authentication Token

Responses

Code	Description
200	Followings List is successfully returned
	Example Value Model
	<pre>{ "number_of_pages": 0, "followings": [{ "id": 0, "name": "string", "surname": "string", "profile_photo": "string", "e_mail": "string", "job_id": 0 }] }</pre>
400	Input Format Error
404	Followings List is empty
500	Database Connection Error

DELETE /follow/followings Takes the following_id and authentication token as inputs and deletes corresponding Follow entry

Parameters

Try it out

Name	Description
following_id * required integer (formData)	ID of the unfollowed user
auth_token * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
200	User is unfollowed successfully
400	Input Format Error
404	No follow information found with given authentication token and following_id
500	Database Connection Error

Profile Management

Profile Management Endpoints

GET /profile/front_page

Parameters

Try it out

Name	Description
auth_token * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
200	OK
500	The server does not respond.

POST /profile/jobs Creates a new job

Parameters Try it out

Name	Description
name <small>required</small> string (formData)	Name of the job
auth_token <small>required</small> string (header)	Authentication Token

Responses Response content type application/json

Code	Description
201	<i>Successfully Created</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

GET /profile/jobs Returns all job names

Parameters Try it out

No parameters

Responses Response content type application/json

Code	Description
200	<i>Valid Response</i>
404	<i>Empty List</i>
500	<i>Database Connection Problem</i>

DELETE /profile/jobs Deletes a job

Parameters

Name	Description
id * required integer (formData)	Job ID
auth_token * required string (header)	Authentication Token

Responses

Code	Description
200	<i>Successfully Deleted</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

Response content type application/json

Try it out

PUT /profile/jobs Updates a job

Parameters

Name	Description
id * required integer (formData)	Job ID
name * required string (formData)	Name of the job
auth_token * required string (header)	Authentication Token

Responses

Code	Description
201	<i>Successfully Updated</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

Response content type application/json

Try it out

GET /profile/notifications Returns all notifications of the logged in user with related user lists

Parameters

Name	Description
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token * required string (header)	Authentication Token

Responses

Response content type: application/json

Code	Description
200	<i>Valid Response</i>
	Example Value: Model
	<pre>{ "number_of_pages": 0, "notification_list": [{ "id": 0, "text": "string", "link": "string", "timestamp": "2021-01-05T19:14:28.231Z", "related_users": ["string"] }] }</pre>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

DELETE /profile/notifications Deletes the given notification

Parameters

Try it out

Name	Description
notification_id * required integer (formData)	ID of the Notification that will be deleted
auth_token * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
200	<i>Successfully Deleted</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
404	<i>Notification Not Found</i>
500	<i>Database Connection Problem</i>

POST /profile/research_information Adds new Research Information

Parameters

Try it out

Name	Description
research_title * required string (formData)	Title of the research
description string (formData)	Description of the work <i>Default value :</i>
year * required integer (formData)	Year of the work
auth_token * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
201	<i>Successfully Created</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

GET /profile/research_information Returns all research information of a user

Parameters

Name	Description
user_id * required integer (query)	User id of requested user
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token * required string (header)	Token that sent via email as a URL link

Responses

Response content type application/json

Code	Description
200	Valid Response Example Value Model <pre>{ "number_of_pages": 0, "research_info": [{ "id": 0, "title": "string", "description": "string", "year": 0 }] }</pre>
400	Wrong Input Format
401	Authentication Problem
403	Private Account Problem
500	Database Connection Problem

DELETE /profile/research_information Deletes Research Information

Parameters Try it out

Name	Description
research_id * required integer (formData)	Research ID of the Research
auth_token * required string (header)	Authentication Token

Responses Response content type application/json

Code	Description
200	<i>Successfully Deleted</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

PUT /profile/research_information Updates a Research Information

Parameters Try it out

Name	Description
research_id * required integer (formData)	Research ID of the Research
research_title string (formData)	Title of the research <i>Default value :</i>
description string (formData)	Description of the work <i>Default value :</i>
year integer (formData)	Year of the work <i>Default value :</i>
auth_token * required string (header)	Authentication Token

Responses Response content type application/json

Code	Description
201	<i>Successfully Updated</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

POST /profile/skills Creates a skill

Parameters

Name	Description
<code>name</code> <small>* required</small> string (formData)	Name of the skill
<code>auth_token</code> <small>* required</small> string (header)	Authentication Token

Responses

Code	Description
201	<i>Successfully Created</i>
400	<i>Wrong Input Format</i>
401	<i>Authentication Problem</i>
500	<i>Database Connection Problem</i>

Try it out

GET /profile/skills Returns all skill names

Parameters

Name	Description
No parameters	

Responses

Code	Description
200	<i>Valid Response</i>
404	<i>Empty List</i>
500	<i>Database Connection Problem</i>

Response content type application/json

Try it out

DELETE /profile/skills Deletes a skill

Parameters

Name	Description
<code>id</code> * required integer (formData)	Skill ID
<code>auth_token</code> * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
200	<code>Successfully Deleted</code>
400	<code>Wrong Input Format</code>
401	<code>Authentication Problem</code>
500	<code>Database Connection Problem</code>

PUT /profile/skills Updates a skill

Parameters

Name	Description
<code>id</code> * required integer (formData)	Skill ID
<code>name</code> * required string (formData)	Name of the skill
<code>auth_token</code> * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
201	<code>Successfully Updated</code>
400	<code>Wrong Input Format</code>
401	<code>Authentication Problem</code>
500	<code>Database Connection Problem</code>

Search Engine Search Engine Endpoints

▼

GET /search_engine/search_history

Parameters

Try it out

Name	Description
search_type * required integer (query)	Search Type 0 => User, 1 => Workspace, 2 => Upcoming Event
auth_token * required string (header)	Authentication Token(if registered)

Responses

Response content type application/json

Code	Description
200	Valid Search History Example Value Model <pre>{ "search_history": { "query": "string", "number_of_use": 0 } }</pre>
400	Input Format Error
401	Account Problems
500	Database Connection Error

GET /search_engine/upcoming_events

Parameters

Name	Description
search_query <small>required</small> string (query)	Search Query
date_filter_start string (query)	Inclusive.
date_filter_end string (query)	Exclusive.
deadline_filter_start string (query)	Inclusive.
deadline_filter_end string (query)	Exclusive.
sorting_criteria integer (query)	None => Semantic Rating // 0 => Alphabetical Order(A>Z) // 1 => Date Order
page integer (query)	Page ID(0-Indexed)
per_page integer (query)	Number of Records in a Page
auth_token string (header)	Authentication Token(if registered)

Responses

Response content type: application/json

Code	Description
200	Valid Search Result
	Example Value Model
	{ "number_of_pages": 0, "upcoming_events": [{ "id": 0, "acronym": "string", "title": "string", "location": "string", "link": "string", "date": "string", "deadline": "string" }] }
400	Input Format Error
401	Account Problems
500	Database Connection Error

GET /search_engine/user

Parameters

Name	Description
search_query * required string (query)	Search Query
job_filter integer (query)	Job Filter(Give as ID)
Sorting_criteria integer (query)	None => Semantic Rating // 0 => Alphabetical Order(A=>Z) // 1 => Alphabetical Order (Z=>A)
page integer (query)	Page ID(0-Indexed)
per_page integer (query)	Number of Records in a Page
auth_token string (header)	Authentication Token(if registered)

Responses

Response content type: application/json

Code	Description
200	Valid Search Result
	Example Value Model
	{ "number_of_pages": 0, "result_list": [{ "id": 0, "name": "string", "surname": "string", "profile_photo": "string", "e_mail": "string", "job_id": 0 }] }
400	Input Format Error
401	Account Problems
500	Database Connection Error

GET /search_engine/workspace

Parameters

Name	Description
search_query * required	Search Query string (query)
skill_filter	Skill Filter string (query)
Creator_name	Name of the creator string (query)
creator_surname	Surname of the creator string (query)
starting_date_start	Inclusive. string (query)
Starting_date_end	Exclusive. string (query)
deadline_start	Inclusive. string (query)
deadline_end	Exclusive. string (query)
Sorting_criteria	None. Semantic Rating: 0: Ascending Date, 1: Descending Date, 2: Ascending Number of Collaborators Needed, 3: Descending Number of Collaborators Needed, 4: Ascending Alphabetical Order, 5: Descending Alphabetical Order integer (query)
page	Page ID(0-Indexed) integer (query)
per_page	Number of Records in a Page integer (query)
auth_token	Authentication Token(if registered) string (header)

Responses

Response content type: application/json

Code	Description
200	Valid Search Result
	Example Value Model
	{ "workspaces": [{ "id": 0, "title": "string", "description": "string", "state": 0, "creator_id": 0, "contributors": [{ "id": 0, "name": "string", "surname": "string" }] }] }
400	Input Format Error
401	Account Problems
500	Database Connection Error

Workspace System Workspace System Endpoints

POST /workspaces Creates a new workspace

Parameters

Try it out

Name	Description
title * required string (formData)	Title of the new workspace
description * required string (formData)	Description of the new workspace
is_private integer (formData)	Privacy status of the new workspace
max_collaborators integer (formData)	Maximum number of collaborators of the new workspace
deadline string (formData)	Deadline of the new workspace
requirements string (formData)	The list of the requirements to be able to join the new workspace
Skills string (formData)	The list of the skills required to be able to join the new workspace
upcoming_events string (formData)	The list of the upcoming events that are related to the new workspace
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
201	Workspace has been successfully created.
400	Missing data fields or invalid data.
500	The server is not connected to the database.

GET /workspaces Returns the requested workspace(s)

Parameters Try it out

Name	Description
workspace_id * required integer (query)	ID of the requested workspace
auth_token * required string (header)	Authentication token

Responses Response content type application/json

Code	Description
200	Workspace(s) found.
400	Missing data fields or invalid data.
401	The user is not allowed to view this workspace.
404	Requested workspace is not found.
500	The server is not connected to the database.

DELETE /workspaces Deletes the requested workspace

Parameters Try it out

Name	Description
workspace_id * required integer (formData)	ID of the requested workspace
auth_token * required string (header)	Authentication token

Responses Response content type application/json

Code	Description
200	Workspace has been successfully deleted.
400	Missing data fields or invalid data.
401	You are not the creator of this workspace, you cannot delete it.
404	The workspace is not found.
500	The server is not connected to the database.

PUT /workspaces Updates the requested workspace

Parameters	
Name	Description
workspace_id * required integer (formData)	ID of the workspace to be updated
title string (formData)	Updated title of the new workspace
description string (formData)	Updated description of the new workspace
is_private integer (formData)	Updated privacy status of the new workspace
max_collaborators integer (formData)	Updated maximum number of collaborators of the new workspace
deadline string (formData)	Updated deadline of the new workspace
requirements string (formData)	Updated list of the requirements to be able to join the new workspace
skills string (formData)	Updated list of the skills required to be able to join the new workspace
state integer (formData)	Updated state of the workspace
upcoming_events string (formData)	The list of the upcoming events that are related to the new workspace
auth_token * required string (header)	Authentication token

Responses	
Code	Description
200	<i>Workspace has been successfully updated.</i>
202	<i>Server has received the request but there was no information to be updated.</i>
400	<i>Missing data fields or invalid data.</i>
401	<i>You are not among the active contributors of this workspace, you cannot modify it.</i>
404	<i>The workspace is not found.</i>
500	<i>The server is not connected to the database.</i>

POST /workspaces/applications Creates an application for a workspace

Parameters

Try it out

Name	Description
workspace_id * required integer (formData)	ID of the workspace that the application is sent to
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
201	<i>Application has been successfully created.</i>
400	<i>Missing data fields or invalid data.</i>
404	<i>The workspace does not exist.</i>
409	<i>The user is already an active contributor of this workspace OR there is already an application sent to this workspace. OR the workspace has already reached its maximum number of collaborators OR this workspace is not in 'search for collaborators' state.</i>
500	<i>The server is not connected to the database.</i>

GET /workspaces/applications From the database, retrieves the requested pending application(s) sent to the workspace and returns them

Parameters

Try it out

Name	Description
application_id integer (query)	ID of the application to be retrieved from the database
workspace_id * required integer (query)	ID of the workspace that the application(s) is sent to
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
200	<i>Application(s) found.</i>
400	<i>Missing data fields or invalid data.</i>
403	<i>The user is not an active contributor of this workspace.</i>
404	<i>Application(s) not found.</i>
500	<i>The server is not connected to the database.</i>

DELETE /workspaces/applications Responds to the application with the given ID

Parameters

Name	Description
<code>application_id</code> * required integer (formData)	ID of the application sent
<code>is_accepted</code> * required string (formData)	Acceptance status of the collaboration application
<code>auth_token</code> * required string (header)	Authentication token

Responses

Response content type: application/json

Code	Description
200	<code>Application has been successfully responded.</code>
400	<code>Missing data fields or invalid data.</code>
403	<code>The user is not an active contributor of this workspace.</code>
404	<code>The application is not found.</code>
500	<code>The server is not connected to the database.</code>

POST /workspaces/invitations Creates an invitation for a workspace
 (i.e. an invitation for a workspace gets sent to the invitee user)

Parameters

Name	Description
workspace_id * required integer (formData)	ID of the workspace that the invitation is sent for
invitee_id * required integer (formData)	ID of the user that the invitation is sent to
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
201	<i>Invitation has been successfully created.</i>
400	<i>Missing data fields or invalid data.</i>
401	<i>You are not the creator of this workspace, you cannot invite users to this workspace.</i>
404	<i>The workspace or the invitee user does not exist.</i>
409	<i>Invitee user is already an active contributor of this workspace OR there is already an invitation sent to the invitee user for this workspace. OR the workspace has already reached its maximum number of collaborators OR this workspace is not in 'search for collaborators' state.</i>
500	<i>The server is not connected to the database.</i>

GET /workspaces/invitations From the database, retrieves the requested pending invitation(s) sent to the user and returns them

Parameters

Name	Description
invitation_id	ID of the invitation to be retrieved from the database
auth_token	Authentication token

Responses

Response content type application/json

Code	Description
200	<i>Invitation(s) found.</i>
400	<i>Missing data fields or invalid data.</i>
404	<i>Invitation(s) not found.</i>
500	<i>The server is not connected to the database.</i>

DELETE /workspaces/invitations Responds to the invitation with the given ID

Parameters Try it out

Name	Description
Invitation_Id * required integer (formData)	ID of the invitation sent
is_accepted * required string (formData)	Acceptance status of the collaboration invitation
auth_token * required string (header)	Authentication token

Responses Response content type application/json

Code	Description
200	<i>Invitation has been successfully responded.</i>
400	<i>Missing data fields or invalid data.</i>
401	<i>You are not the invitee of this invitation, you cannot respond to it.</i>
404	<i>The invitation is not found.</i>
500	<i>The server is not connected to the database.</i>

POST /workspaces/issue Creates issue

Parameters

Name **Description**

workspace_id * required
integer
(formData)

title * required
string
(formData)

description * required
string
(formData)

deadline
string
(formData)

auth_token * required
string
(header)

Responses

Response content type **application/json**

Code **Description**

200 **Success**

Example Value | Model

```
{ "msg": "string", "issue_id": 0, "workspace_id": 0, "title": "string", "description": "string", "deadline": "2021-01-05T19:14:28.272Z", "is_open": true, "creator_id": 0 }
```

400 **Input Format Error**

401 **Account Problems**

404 **Not found**

500 **Database Connection Error**

GET /workspaces/{workspace_id}/issue GetIssues

Parameters

Name **Description**

workspace_id * required integer (query)	ID of the requested workspace
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token * required string (header)	Authentication token

Responses

Response content type: application/json

Code	Description
200	Success
	Example Value Model
	{ "number_of_pages": 0, "result": [{ "issue_id": 0, "workspace_id": 0, "title": "string", "description": "string", "deadline": "2021-01-05T19:14:28.274Z", "is_open": true, "creator_name": "string", "creator_surname": "string", "creator_e_mail": "string", "creator_rate": 0, "creator_job_name": "string", "creator_institution": "string", "creator_is_private": true, "creator_photo": "string" }] }
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

DELETE /workspaces/issue Deletes an issue

Parameters

Name	Description
workspace_id * required integer (formData)	ID of the requested workspace
issue_id * required integer (formData)	ID of the issue
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

PUT /workspaces/issue Updates an issue

Parameters

Name	Description
workspace_id * required integer (formData)	ID of the requested workspace
issue_id * required integer (formData)	ID of the issue
title string (formData)	Title of the updated issue
description string (formData)	Description of the updated issue
deadline string (formData)	Updated deadline of the new issue
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

POST /workspaces/{issue/assignee} Create issue assignee record

Parameters

Name **Description**

workspace_id * required integer (<i>formData</i>)	ID of the requested workspace
issue_id * required integer (<i>formData</i>)	ID of the issue
assignee_id * required integer (<i>formData</i>)	ID of the assignee
auth_token * required string (<i>header</i>)	Authentication token

Responses Response content type **application/json**

Code	Description
400	<i>Input Format Error</i>
401	<i>Account Problems</i>
404	<i>Not found</i>
500	<i>Database Connection Error</i>

GET /workspaces/{workspace_id}/issue/{issue_id}/assignee Get Issue Assignees

Parameters

Name	Description
workspace_id * required integer (query)	ID of the requested workspace
issue_id * required integer (query)	ID of the issue
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token * required string (header)	Authentication token

Responses

Response content type **application/json**

Code	Description
200	Success
	Example Value Model
	{ "number_of_pages": 0, "result": [{ "comment_id": 0, "comment": "string", "owner_id": 0, "owner_name": "string", "owner_surname": "string", "owner_e_mail": "string", "owner_rate": 0, "owner_photo": "string" }] }
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

DELETE /workspaces/issue/assignee Deletes an issue assignee

Parameters

Name	Description
<code>workspace_id</code> * required integer (formData)	ID of the requested workspace
<code>issue_id</code> * required integer (formData)	ID of the issue
<code>assignee_id</code> * required integer (formData)	ID of the assignee
<code>auth_token</code> * required string (header)	Authentication token

Responses

Code	Description	Response content type
400	<code>Input Format Error</code>	application/json
401	<code>Account Problems</code>	
404	<code>Not found</code>	
500	<code>Database Connection Error</code>	

POST /workspaces/issue/comment Create issue comment

Parameters

Name	Description
<code>workspace_id</code> * required integer (formData)	ID of the requested workspace
<code>issue_id</code> * required integer (formData)	ID of the issue
<code>comment</code> * required string (formData)	Comment
<code>auth_token</code> * required string (header)	Authentication token

Responses

Code	Description	Response content type
400	<code>Input Format Error</code>	application/json
401	<code>Account Problems</code>	
404	<code>Not found</code>	
500	<code>Database Connection Error</code>	

GET /workspaces/{workspace_id}/issue/{issue_id}/comment Get Issue Comments

Parameters

Name	Description
workspace_id * required integer (query)	ID of the requested workspace
issue_id * required integer (query)	ID of the issue
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token * required string (header)	Authentication token

Responses

Response content type: application/json

Code	Description
200	Success
	Example Value Model
	{ "number_of_pages": 0, "result": [{ "comment_id": 0, "comment": "string", "owner_id": 0, "owner_name": "string", "owner_surname": "string", "owner_e_mail": "string", "owner_rate": 0, "owner_photo": "string" }] }
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

DELETE /workspaces/issue/comment Deletes an issue comment

Parameters	
Name	Description
workspace_id * required integer (formData)	ID of the requested workspace
issue_id * required integer (formData)	ID of the issue
comment_id * required integer (formData)	ID of the comment
auth_token * required string (header)	Authentication token

Responses

		Response content type
		application/json
Code	Description	
400	<i>Input Format Error</i>	
401	<i>Account Problems</i>	
404	<i>Not found</i>	
500	<i>Database Connection Error</i>	

POST /workspaces/milestone Creates milestone

Parameters	
Name	Description
workspace_id * required integer (formData)	ID of the requested workspace
title * required string (formData)	Title of the new milestone
description * required string (formData)	Description of the new milestone
deadline * required string (formData)	Should be in DateTime form. e.g. 2015-12-20 10:01:00
auth_token * required string (header)	Authentication token

Responses

		Response content type
		application/json
Code	Description	
400	<i>Input Format Error</i>	
401	<i>Account Problems</i>	
404	<i>Not found</i>	
500	<i>Database Connection Error</i>	

GET /workspaces/milestone Get Milestones

Parameters

Try it out

Name	Description
workspace_id <small>* required</small> integer (query)	ID of the requested workspace
page integer (query)	Page index that you want(Starts from 0)
per_page integer (query)	Number of items in a page
auth_token <small>* required</small> string (header)	Authentication token

Responses

Response content type application/json

Code	Description
200	Success
	Example Value Model
	<pre>{ "number_of_pages": 0, "result": [{ "milestone_id": 0, "workspace_id": 0, "title": "string", "description": "string", "deadline": "2021-01-05T19:14:28.288Z", "creator_id": 0, "creator_name": "string", "creator_surname": "string", "creator_e_mail": "string", "creator_rate": 0, "creator_job_name": "string", "creator_institution": "string", "creator_is_private": true }] }</pre>
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

DELETE /workspaces/milestone Deletes a milestone

Parameters

Try it out

Name	Description
workspace_id * required integer (formData)	ID of the requested workspace
milestone_id * required integer (formData)	ID of the milestone
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

PUT /workspaces/milestone Updates a milestone

Parameters

Try it out

Name	Description
workspace_id * required integer (formData)	ID of the requested workspace
milestone_id * required integer (formData)	ID of the issue
title string (formData)	Title of the updated milestone
description string (formData)	Description of the updated milestone
deadline string (formData)	Updated deadline of the milestone
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

DELETE /workspaces/quit Use it to quit from a workspace

Parameters

Try it out

Name	Description
<code>workspace_id</code> * required integer (formData)	ID of the Workspace
<code>auth_token</code> * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
201	<code>You successfully quited from workspace</code>
400	<code>Missing or invalid data</code>
401	<code>Login Required</code>

GET /workspaces/self Returns all workspaces of the user(Public and Private)

Parameters

Try it out

Name	Description
<code>auth_token</code> * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
200	<code>Valid Response</code>
401	<code>Authentication Problem</code>
500	<code>Database Connection Problem</code>

Example Value | Model

```
{
  "workspaces": [
    {
      "id": 0,
      "title": "string",
      "description": "string",
      "state": 0,
      "creator_id": 0,
      "contributors": [
        {
          "id": 0,
          "name": "string",
          "surname": "string"
        }
      ]
    }
  ]
}
```

GET /workspaces/trending_projects

Parameters

Name Description

number_of_workspaces * required
integer
(query) Number of Workspaces

Responses

Code Description Response content type

200 Valid Response application/json

Example Value | Model

```
{ "trending_projects": [ { "id": 0, "title": "string", "description": "string", "state": 0, "creator_id": 0, "contributors": [ { "id": 0, "name": "string", "surname": "string" } ] } ] }
```

400 Invalid Input Format

500 Database Connection Error

GET /workspaces/user Only returns public workspaces of the user with given ID

Parameters

Name	Description
<code>user_id</code> <small>* required</small> integer (query)	ID of the user
<code>auth_token</code> <small>* required</small> string (header)	Authentication token

Responses

Response content type `application/json`

Code	Description
200	<code>Valid Response</code> Example Value Model <pre>{ "workspaces": [{ "id": 0, "title": "string", "description": "string", "state": 0, "creator_id": 0, "contributors": [{ "id": 0, "name": "string", "surname": "string" }] }] }</pre>
400	<code>Invalid Input</code>
401	<code>Authentication Problem</code>
403	<code>Private Profile</code>
500	<code>Database Connection Problem</code>

File System File System Endpoints

POST /file_system/file

Parameters

Try it out

Name	Description
workspace_id * required integer (formData)	Workspace ID
path * required string (formData)	Path of the File
filename * required string (formData)	Name of the File
new_file * required file (formData)	Upload File Here
auth_token * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
201	<i>File Successfully Uploaded</i>
400	<i>Invalid Input</i>
401	<i>Authentication Problem</i>
403	<i>Forbidden Path</i>
404	<i>User if not found</i>

GET /file_system/file

Parameters

Name	Description
workspace_id * required integer (query)	Workspace ID
path * required string (query)	Path of the File
filename * required string (query)	Name of the File
auth_token * required string (header)	Authentication Token

Responses

Code	Description	Response content type
200	Valid Data	application/json
400	Invalid Input	
401	Authentication Problem	
403	Forbidden Path	
404	User if not found	

DELETE /file_system/file

Parameters

Name	Description
workspace_id * required integer (formData)	Workspace ID
path * required string (formData)	Path of the File
filename * required string (formData)	Name of the File
auth_token * required string (header)	Authentication Token

Responses

Code	Description	Response content type
200	File Successfully Deleted	application/json
400	Invalid Input	
401	Authentication Problem	
403	Forbidden Path	
404	User if not found	

PUT /file_system/file

Parameters

Try it out

Name	Description
<code>workspace_id</code> * required integer (formData)	Workspace ID
<code>path</code> * required string (formData)	Path of the File
<code>filename</code> * required string (formData)	Name of the File
<code>new_file</code> * required file (formData)	Upload File Here
<code>auth_token</code> * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
200	<code>File Name Successfully Changed</code>
400	<code>Invalid Input</code>
401	<code>Authentication Problem</code>
403	<code>Forbidden Path</code>
404	<code>User if not found</code>

POST /file_system/folder

Parameters

Try it out

Name	Description
workspace_id * required integer (formData)	Workspace ID
path * required string (formData)	Path of the Folder
new_folder_name * required string (formData)	Name of the new folder
auth_token * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
201	<i>Folder Successfully Created</i>
400	<i>Invalid Input</i>
401	<i>Authentication Problem</i>
403	<i>Forbidden Path</i>
404	<i>User if not found</i>

GET /file_system/folder

Parameters

Name	Description
workspace_id <small>* required</small> integer (query)	Workspace ID
path <small>* required</small> string (query)	Path of the Folder
auth_token <small>* required</small> string (header)	Authentication Token

Responses

Code	Description	Response content type
200	Valid Folder	application/json
	Example Value Model	
	{ "files": ["string"], "folders": ["string"], "cwd": "string" }	
400	Invalid Input	
401	Authentication Problem	
403	Forbidden Path	
404	User if not found	

DELETE /file_system/folder

Parameters

Name	Description
workspace_id <small>* required</small> integer (FormData)	Workspace ID
path <small>* required</small> string (FormData)	Path of the Folder
auth_token <small>* required</small> string (header)	Authentication Token

Responses

Code	Description	Response content type
200	Folder Successfully Deleted	application/json
400	Invalid Input	
401	Authentication Problem	
403	Forbidden Path	
404	User if not found	

PUT /file_system/folder

Parameters

Name	Description
workspace_id <small>* required</small> integer (formData)	Workspace ID
path <small>* required</small> string (formData)	Path of the Folder
new_folder_name <small>* required</small> string (formData)	Name of the new folder
auth_token <small>* required</small> string (header)	Authentication Token

Responses

Code	Description	Response content type
200	Folder Name Successfully Changed	application/json
400	Invalid Input	
401	Authentication Problem	
403	Forbidden Path	
404	User if not found	

Upcoming Events

Upcoming Events Endpoints

GET /upcoming_events

Parameters

Name	Description
page integer (query)	Page ID(0-Indexed)
per_page integer (query)	Number of Records in a Page

Responses

Code	Description	Response content type
200	Valid Response	application/json
500	Database Connection Problem	

Example Value | Model

```
{
  "number_of_pages": 0,
  "upcoming_events": [
    {
      "id": 0,
      "acronym": "string",
      "title": "string",
      "location": "string",
      "link": "string",
      "date": "string",
      "deadline": "string"
    }
  ]
}
```

GET

/auth_system/admin Deletes the specified user from the system

Parameters

[Try it out](#)

No parameters

Responses

Response content type [application/json](#) ▾

Code	Description
------	-------------

200	<i>Valid Response</i>
-----	-----------------------

400	<i>Inappropriate Input</i>
-----	----------------------------

401	<i>Unauthorized Input</i>
-----	---------------------------

500	<i>Database Connection Error</i>
-----	----------------------------------

POST /auth_system/admin Suspends the specified user from the system

Parameters

No parameters

Responses

Response content type application/json

Code	Description
200	Valid Response
400	Inappropriate Input
401	Unauthorized Input
500	Database Connection Error

GET /follow/report Get Reports of User

Parameters

Name Description

page
integer
(query)

per_page
integer
(query)

auth_token * required
string
(header)

Responses

Response content type application/json

Code	Description
200	Success

Example Value | Model

```
{
  "number_of_pages": 0,
  "result": [
    {
      "report_id": 0,
      "owner_id": 0,
      "reported_user_id": 0,
      "reported_user_name": "string",
      "reported_user_surname": "string",
      "text": "string"
    }
  ]
}
```

400 **Input Format Error**

401 **Account Problems**

404 **Not found**

500 **Database Connection Error**

DELETE /follow/report Deletes an user report

Parameters	Description	Try it out
report_id * required integer <code>(query)</code>	ID of the report	
auth_token * required string <code>(header)</code>	Authentication token	

Responses Response content type **application/json** ▾

Code	Description
200	Success
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

POST /follow/report Create user report

Parameters

Try it out

Name	Description
reported_user_id * required integer (formData)	ID of the reported user
text string (formData)	Report for the User
auth_token * required string (header)	Authentication token

Responses

Response content type **application/json** ▾

Code	Description
200	Success
400	Input Format Error
401	Account Problems
404	Not found
500	Database Connection Error

GET
/search_engine/tag_search

Parameters
Try it out

Name	Description
search_type integer (query)	0 => User 1 => Worksapce
skills string (query)	The list of the skills required to be able to join the new workspace
page integer (query)	Page ID(0-Indexed)
per_page integer (query)	Number of Records in a Page

Responses
Response content type

application/json

Code	Description
200	<p><i>Valid Users</i></p>
	<p>Example Value Model</p> <pre>{ "number_of_pages": 0, "result_list": [{ "id": 0, "name": "string", "surname": "string", "profile_photo": "string", "e_mail": "string", "job_id": 0 }] }</pre>
201	<p><i>Valid Workspaces</i></p>
	<p>Example Value Model</p> <pre>{ "workspaces": [{ "id": 0, "title": "string", "description": "string", "state": 0, "creator_id": 0, "contributors": [{ "id": 0, "name": "string", "surname": "string" }] }] }</pre>
400	<p><i>Input Format Error</i></p>
401	<p><i>Account Problems</i></p>
500	<p><i>Database Connection Error</i></p>

GET /upcoming_events/personal_calendar Returns the the personal calendar (workspaces, issues, milestones with their deadlines) of a user

Parameters

Try it out

Name	Description
auth_token * required string (header)	Authentication token

Responses

Response content type application/json

Code	Description
200	<i>Calendar has been successfully returned.</i>
500	<i>The server is not connected to the database.</i>

GET /activity_stream Returns activity stream of an user

Parameters

Try it out

Name	Description
page * required integer (query)	Page index that you want(Starts from 0)
per_page * required integer (query)	Number of items in a page
auth_token * required string (header)	Authentication Token

Responses

Response content type application/json

Code	Description
200	<i>Success</i>

[Example Value](#) | [Model](#)

```
{  
  "@context": "string",  
  "summary": "string",  
  "type": "string",  
  "id": 0,  
  "totalItems": 0,  
  "orderedItems": [  
    {  
      "@context": {  
        "@vocab": "string",  
        "@language": "string"  
      },  
      "summary": "string",  
      "type": "string",  
      "actor": {  
        "type": "string",  
        "id": 0,  
        "name": "string",  
        "image": {  
          "type": "string",  
          "url": "string"  
        }  
      },  
      "object": {  
        "type": "string",  
        "id": 0,  
        "name": "string",  
        "image": {  
          "type": "string",  
          "url": "string"  
        }  
      }  
    }  
  ]  
}
```

400

Input Format Error

401

Account Problems

404

Not found

500

Database Connection Error

```
{  
  "@context": "string",  
  "summary": "string",  
  "type": "string",  
  "id": 0,  
  "totalItems": 0,  
  "orderedItems": [  
    {  
      "@context": {  
        "@vocab": "string",  
        "@language": "string"  
      },  
      "summary": "string",  
      "type": "string",  
      "actor": {  
        "type": "string",  
        "id": 0,  
        "name": "string",  
        "image": {  
          "type": "string",  
          "url": "string"  
        }  
      }  
    }  
  ]  
}
```

```
"object": {
    "type": "string",
    "id": 0,
    "name": "string",
    "image": {
        "type": "string",
        "url": "string"
    },
    "content": "string",
    "ext:ratingValue": 0
},
"target": {
    "type": "string",
    "id": 0,
    "name": "string",
    "image": {
        "type": "string",
        "url": "string"
    }
}
}
]
```

REQUIREMENTS

Glossary

1. Account Information: Name, surname, e-mail and affinities are necessary information. Profile photo, research, ResearchGate, and Google Scholar accounts are optional information.
2. Activity Stream: All activities of followed users.(State changes of their workspaces, a update in the users' workspaces)
3. Admin: A user who is capable of managing the whole application(Owner of the site).
4. Collaborator: Any user working on the same workspace.
5. Deadline and Milestone Mechanism: A mechanism to add deadlines for some tasks of the workspace on a timeline. This also helps users to distribute requirements among the collaborators so that they can keep track of the progress.
6. Editable Files: Markdown files(.md), any code file(.c,.py,.cpp,.java,.js,.m), txt files, .tex files
7. Filter: Founder of the Project, Starting Date, Submission Deadline, Research Area/Topic and labels are filters in the advanced search of the platform.
8. Guest: An unregistered person who uses the application.
9. Issue: Issue is a way to distribute requirements or task of the project/paper. An issue item consists of a description, and the user that creates that issue item. An issue consists of issue items and in addition to those,a title, a creator user(which is the user that creates the first issue item), a deadline and an assignee list.
10. Invisible: Any user and guest (excluding the collaborators of the workspace) don't monitor the contents of the workspace.
11. Label: Related keywords about the workspace indicating the contents of the workspace. Users can search the related keywords, and find related the posts with regards to these labels. E.g. State of the Project, research field of the project, difficulty.
12. Ongoing State: It is the second of the three states of a project. This is the development phase of the project. After finding enough collaborators, the project will go to the second state which is the Ongoing State.
13. Published State: It is the last stage of a project. After milestones are completed with the request of owners of the project, the project will go to Published State.
14. User: A user who is registered to the system.
15. Search for Collaborators State: It is the first of the three states of a project. In this state, the founder of the project waits for requests from other Users and sends invitations to Users.
16. Semantic Search: Semantic search is a data searching technique in which a search query aims to not only find keywords but to determine the intent and contextual meaning of the words a person is using for search.

17. Sorting Criteria: Date, number of collaborators needed.
18. Terms of Use: The legal agreements between a service provider and a person who wants to use that service.
19. Trending Projects: Mostly viewed projects in the platform.
20. Upcoming Event: Upcoming Conferences, Journal Special Issues, Submission Deadlines and CFP (call for papers) documents.
21. Visible: Any user inspects the contents of the workspace.
22. Workspace: A storage area for a group of users to develop their project/paper proposal collaboratively.
23. Workspace Information: Long & short description, title, labels, related upcoming events, number of collaborators needed.
24. Workspace Related Materials: Editable files, plots, PDF files.

1 Functional Requirements

1.1 User Requirements

1.1.1 Guests and Registration

1.1.1.1 Guests shall be able to register by providing account information which includes name, surname, e-mail, affinities and optionally profile photo, past research, ResearchGate, and Google Scholar accounts.

1.1.1.2 Guests should be able to register by using their Google accounts.

1.1.1.3 Guests shall accept the Terms of Use and Privacy Policy in order to register.

1.1.1.4 Guests shall be able to do basic & advanced search with using filters.

1.1.1.5 Guests shall be able to see upcoming events and their details, and project descriptions.

1.1.1.6 Guests should be able to see trending projects.

1.1.2 Login

1.1.2.1 Users shall be able to login by using their e-mail addresses and passwords.

1.1.2.2 Users should be able to login by using their Google accounts.

1.1.2.3 Users shall be able to reset their password if they ever forget it.

1.1.3 User Follow & Rating

1.1.3.1 Users shall be able to follow public accounts.

1.1.3.2 Users shall be able to send requests to follow private accounts.

1.1.3.3 Users with private accounts shall be able to accept/reject follow requests.

1.1.3.4 Users shall be able to comment & rate users that are collaborated with.

1.1.3.5 Users shall be able to report another user in case of an inappropriate behavior or content.

1.1.4 Permissions in Projects

1.1.4.1 Users shall be able to create public/private workspace.

1.1.4.2 Users shall be able to invite users to the workspace at Search for Collaborators State.

1.1.4.3 Users shall be able to send a request for collaboration to the public workspaces at Search for Collaborators State or Ongoing State.

1.1.4.4 Users shall be able to accept/reject workspace invitations.

1.1.4.5 Collaborators of the workspace shall be able to accept/reject join requests to their workspaces.

1.1.4.6 Users shall set the workspace information (which is long & short description, title, labels, related upcoming events, number of collaborators) during the workspace creation phase.

1.1.4.7 Collaborators of a workspace shall be able to upload the related materials such as editable files, plots, PDF files.

1.1.4.8 Collaborators of a workspace should be able to edit editable files.

1.1.4.9 Users shall be able to view the related materials of the public workspaces when the project is in the Published State or the Search for Collaborators State.

1.1.4.10 Collaborators shall be able to inspect the related materials of their private workspace.

1.1.4.11 Collaborators shall be able to leave the workspace.

1.1.4.12 Collaborators shall be able to create, close and reopen issues and assign any collaborator. The definition of the issue can be found under the Glossary part.

1.1.4.13 Collaborators shall be able to create milestones and add them to the project timeline to distribute requirements and tasks, and also to keep track of the progress.

1.1.4.14 Collaborators shall be able to change the state of the workspace.

1.1.4.15 Collaborators shall be able to link the workspace to an upcoming event in the system.

1.1.5 Profile Management

1.1.5.1 Users shall be able to edit their account information which includes name, surname, e-mail, affinities, profile photo, research, ResearchGate and Google Scholar accounts.

1.1.5.2 Users shall be able to set their profiles as either public or private.

1.1.5.3 Users shall be able to link their Google Scholar & ResearchGate accounts.

1.1.5.4 A user shall be able to view public profile pages.

1.1.5.5 Users shall be able to view private profiles that they are following.

1.1.5.6 Users shall be able to mute all of the notifications.

1.1.5.7 Users shall be able to view sections as followers, comments, ratings, photo, name and e-mail in a profile.

1.1.6 Search

1.1.6.1 A user and a guest shall be able to search public workspaces, upcoming events, and users using keywords.

1.1.6.2 Users and guests shall be able to do an advanced search on only public workspaces, upcoming events, and any profile by specifying filters such as founder of the project, starting date, submission deadline, research area/topic and labels.

1.1.6.3 Users and guests should be able to sort search results of workspaces according to the sorting criteria as date, and number of collaborators needed, alphabetical order.

1.1.6.4 Users and guests should be able to sort search results of upcoming events according to the sorting criteria as date, alphabetical order.

1.1.6.5 Users and guests should be able to sort search results of users according to the sorting criteria as alphabetical order.

1.2 System Requirements

1.2.1 System

1.2.1.1 System shall provide the activities of the users that are followed or in collaboration as a stream in the home page.

1.2.1.2 System shall provide date information about upcoming events.

1.2.1.3 System should provide profile recommendations for new people to follow.

1.2.1.4 System shall provide workspace recommendations for Users which are at Search for Collaborators State according to the requirements explained in 1.2.4.

1.2.1.5 System shall verify each newly registered account via e-mail.

1.2.2 Search Engine

1.2.2.1 System shall provide a mechanism that users can use to search.

1.2.2.2 Search engine shall search content concerning the research area, topic, and scope.

1.2.2.3 Search engine shall provide recent search history for the users.

1.2.2.4 System shall support searching for semantically related content.

1.2.2.5 System should have a tag mechanism to enhance the recommendation, searching and related functionalities.

1.2.3 Recommendation

- 1.2.3.1 The platform shall support a recommendation system.
- 1.2.3.2 The platform shall provide definitive labels according to the user's skills.
- 1.2.3.3 Recommendation system shall recommend appropriate projects for the interests and technical suitability to Users.
- 1.2.3.4 Recommendation system shall recommend appropriate Users for the projects which are at the Search for Collaborators State.

1.2.4 Notification

- 1.2.4.1 The system should send notifications to users when new conferences have been added.
- 1.2.4.2 The system shall send notifications when the user has new followers or follow requests.
- 1.2.4.3 The system shall send notifications when a user gets a new comment or rating.
- 1.2.4.4 The system shall send notifications to the user when he/she gets accepted for collaboration from a project.
- 1.2.4.5 The system shall send notifications to the collaborators of the workspace when any application for collaboration request is sent to that workspace.

1.2.5 Project/Issue Management System

- 1.2.5.1 System shall support three different states for the workspaces, namely Search for Collaborators State, Ongoing State and Published State.
- 1.2.5.2 System shall provide a functionality for collaborators to share materials such as PDF files, plots etc. in the workspace.
- 1.2.5.3 System shall prevent users who do not collaborate to access private workspaces

1.2.6 Profile Page

- 1.2.6.1 A profile page shall be public or private.
- 1.2.6.2 Profile Page of any user shall be different from each other at least by e-mail information.
- 1.2.6.3 Profile page shall include the research information which is fetched from Google Scholar or ResearchGate account of the user if these accounts are provided to the system.
- 1.2.6.4 Profile page shall provide the follower information which is the number of followers, the number of followed users and links to profile pages of followers and followed users.
- 1.2.6.5 Profile page shall provide account information which includes name, surname, e-mail certainly and affinities, profile photo, research, ResearchGate and Google Scholar accounts if these information is provided to the system.
- 1.2.6.6 A public profile page and all information provided (described in 1.2.6.3, 1.2.6.4, 1.2.6.5) shall be able to be viewed by any user in the system.

1.2.6.7 A private profile page and all information provided (described in 1.2.6.3, 1.2.6.4, 1.2.6.5) shall be able to be viewed by only the followers of the owner of the profile page.

2. Non-Functional Requirements

2.1 Performance & Reliability

2.1.1 System shall respond in at most 3 seconds.

2.1.2 Search shall give result in at most 5 seconds.

2.1.3 The system should work 7/24 with no more than 1% downtime.

2.2 Security

2.2.1 Passwords shall be encrypted with SHA-256.

2.2.2 System shall use HTTPS Protocol.

2.2.3 System objects shall be encrypted.

2.2.4 System shall be backed up to AWS at the end of each day according to GMT +3.

2.3 Portability

2.3.1 Mobile Application shall support Android version 7.1.2 or later.

2.3.2 Web Application shall support Chrome version 79.0.3945.130 or later.

2.3.3 Web Application shall support Firefox version 73.0.1 or later.

2.3.4 Web Application shall support Safari version 13.0.5 or later.

2.3.5 Web Application shall support Opera version 66.0.3515.115 or later.

2.4 Ethical Issues and Legal Constraints

2.4.1 In this application "personal information", "contact information", "copyrighted contents", etc. shall be used according to the rules of GDPR and KVKK.

2.4.2 In this project, ethical constraints shall be complied according to ACM Code of Ethics and Professional Conduct document.

2.4.3 New users should accept the "Terms of Use" and "Privacy Policy" of the platform which are prepared according to relevant regulations.

2.5 Implementation

2.5.1 The application shall be deployable on a remote and manually configurable remote server.

2.5.2 The application should be dockerized.

2.5.3 The application shall include necessary API implementations.

2.5.4 There shall be a web platform and a native Android application that supports the same functionalities.

2.5.5 The implementation of the system should follow W3C standards and W3C Activity Streams Protocol.

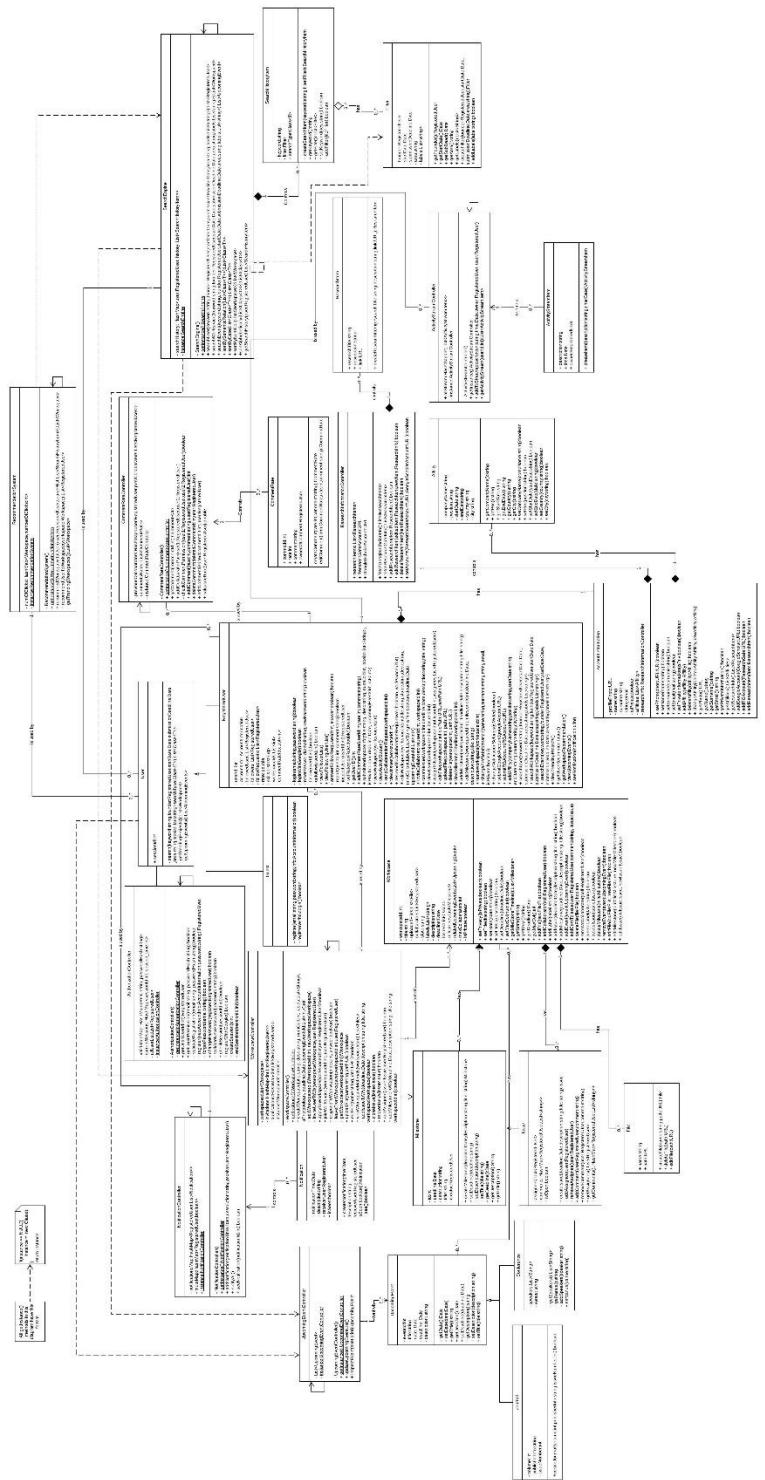
References

https://en.wikipedia.org/wiki/Privacy_policy

https://en.wikipedia.org/wiki/Terms_of_service

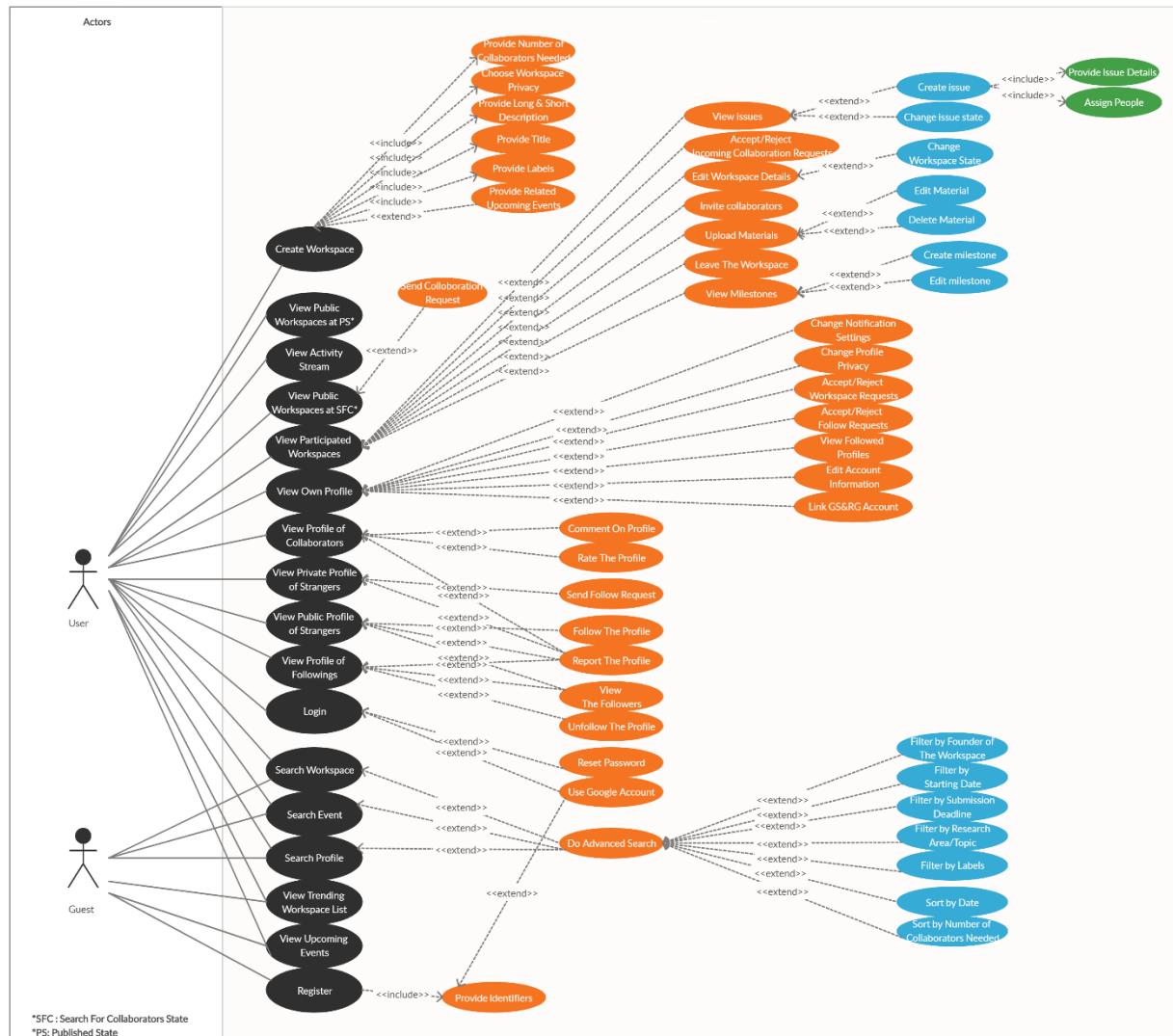
DESIGN DOCUMENTS

CLASS DIAGRAM



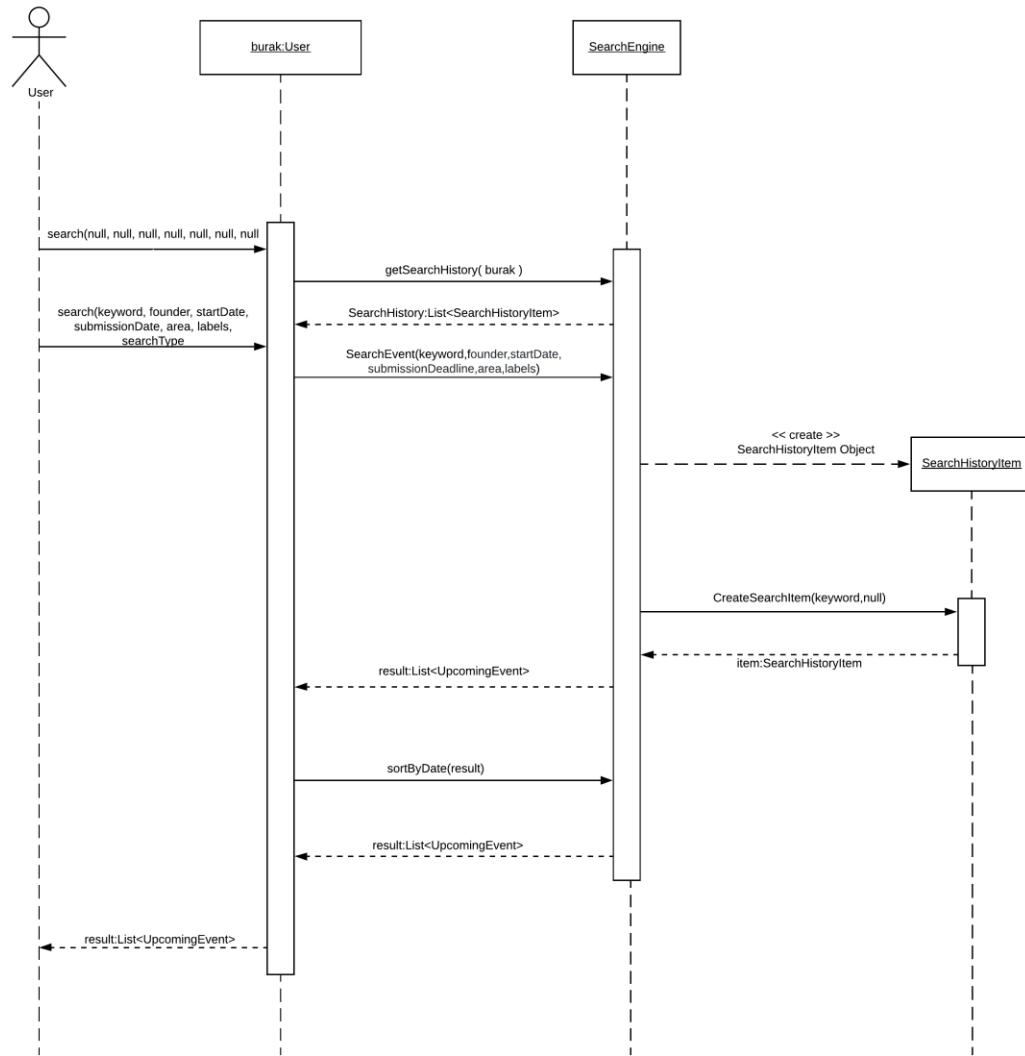
<https://www.lucidchart.com/invitations/accept/d7800851-d1ef-4030-b3c6-980eb0730dc>

USE CASE DIAGRAM

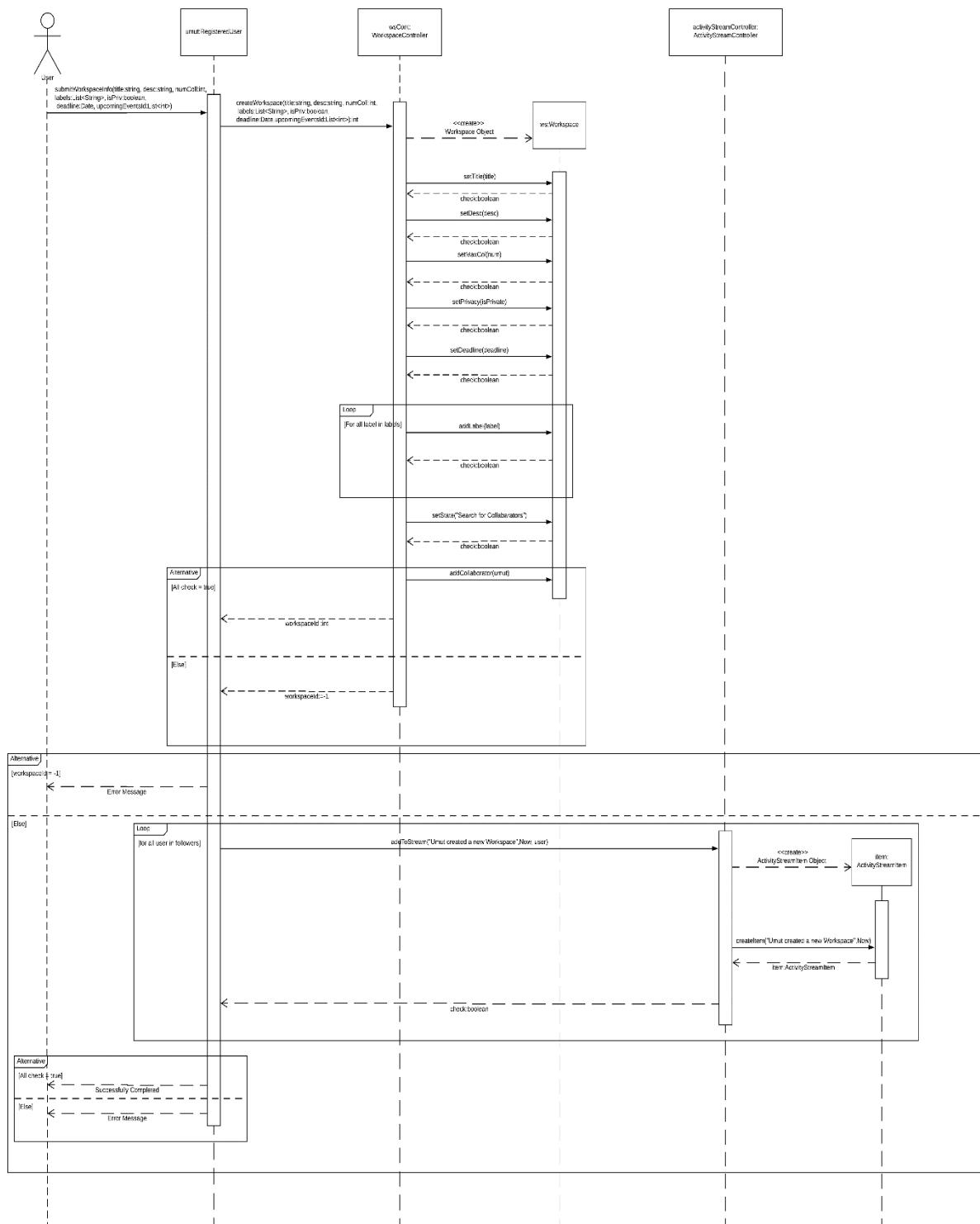


SEQUENCE DIAGRAMS

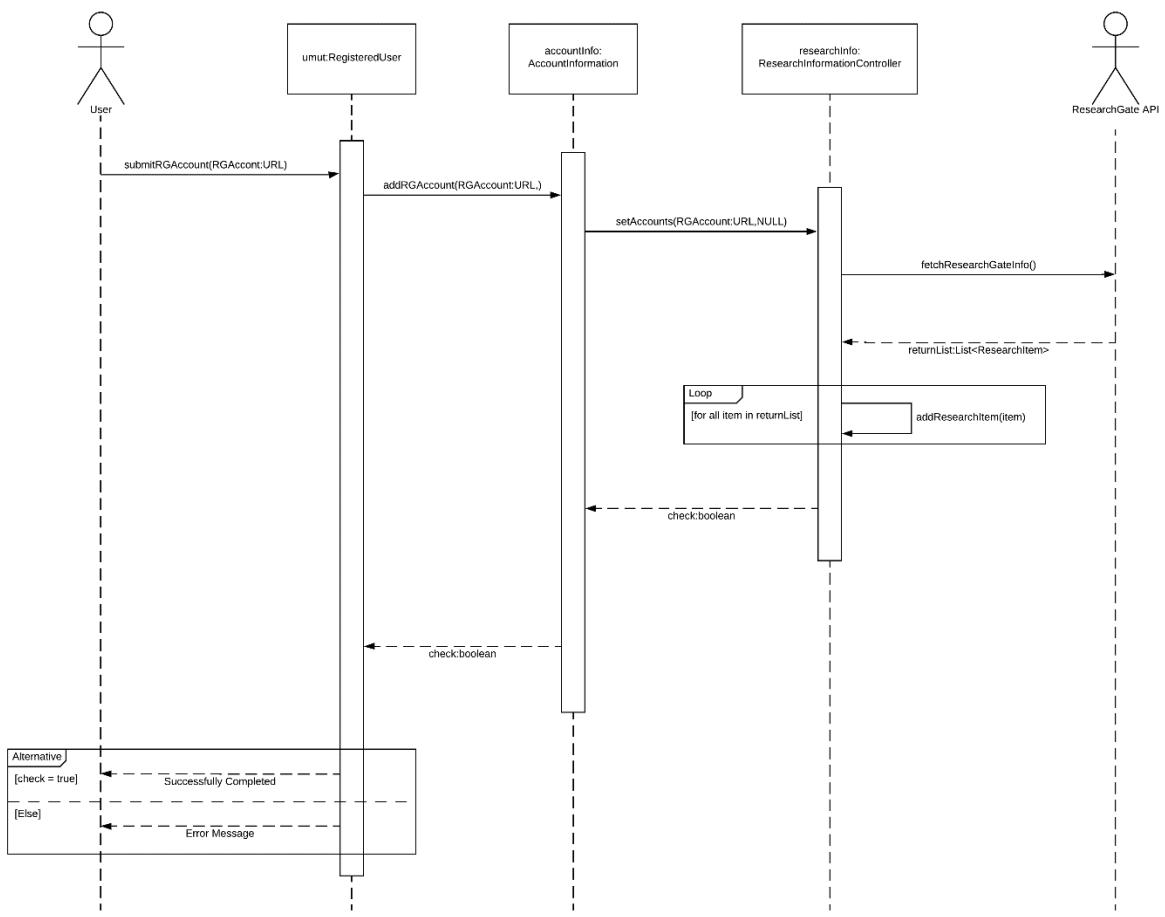
Keyword Search with Date Sorting



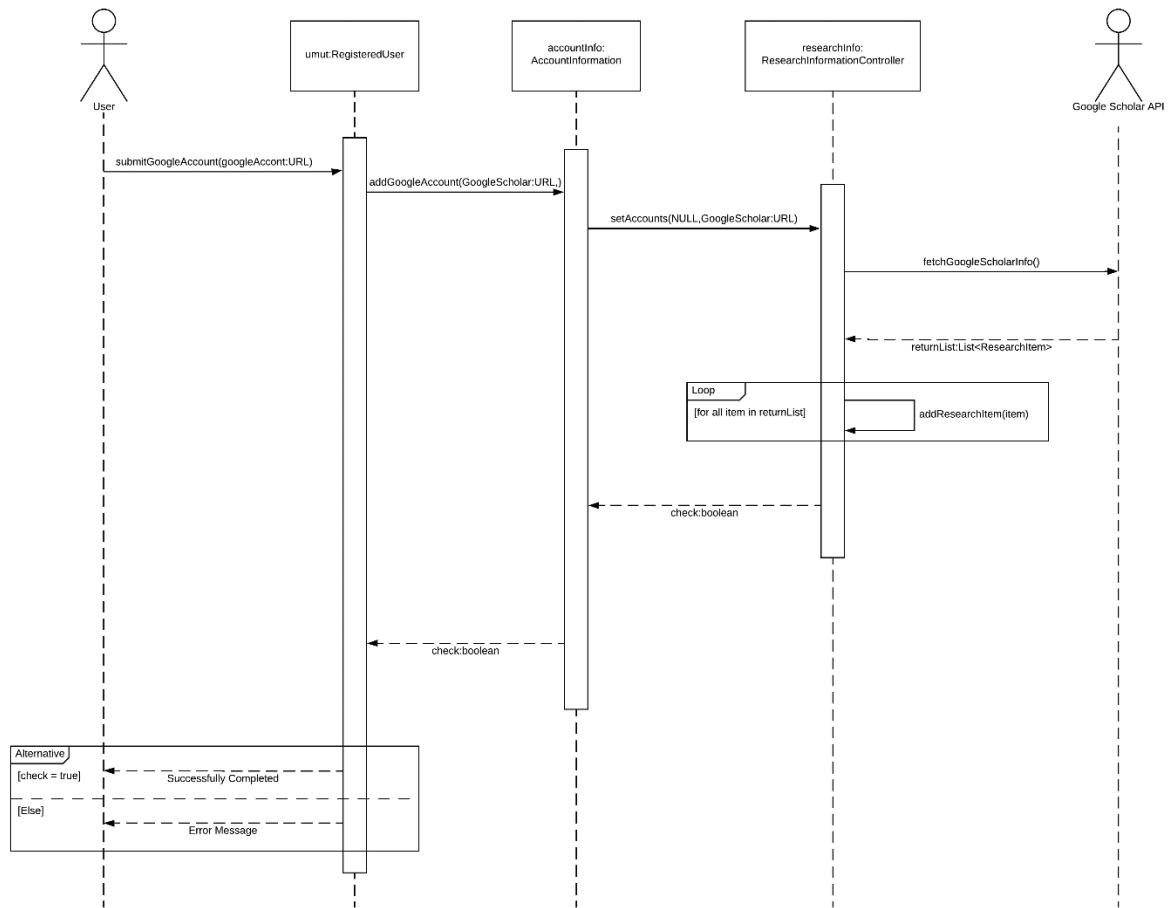
Workspace Creation



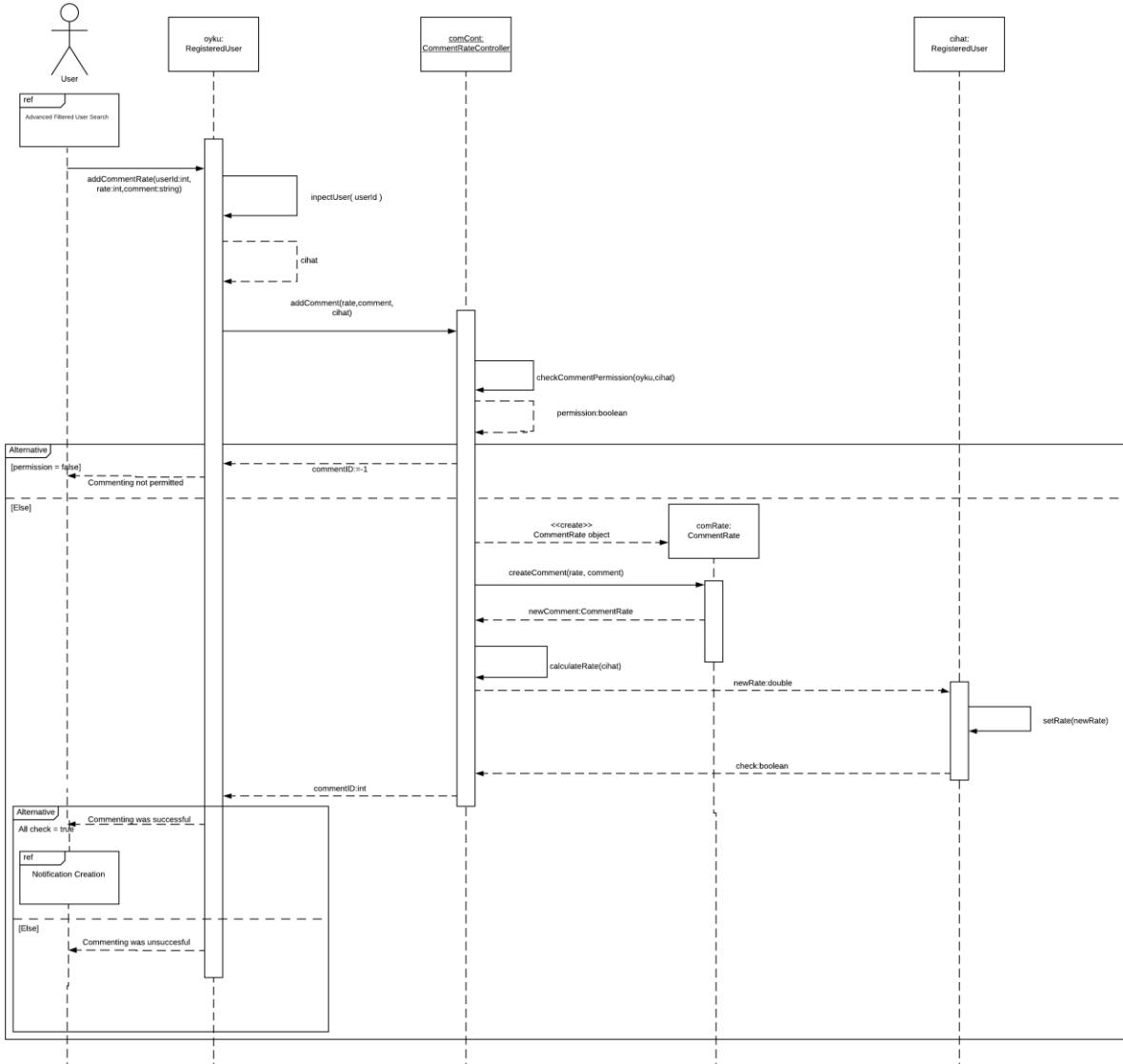
Link ResearchGate Account



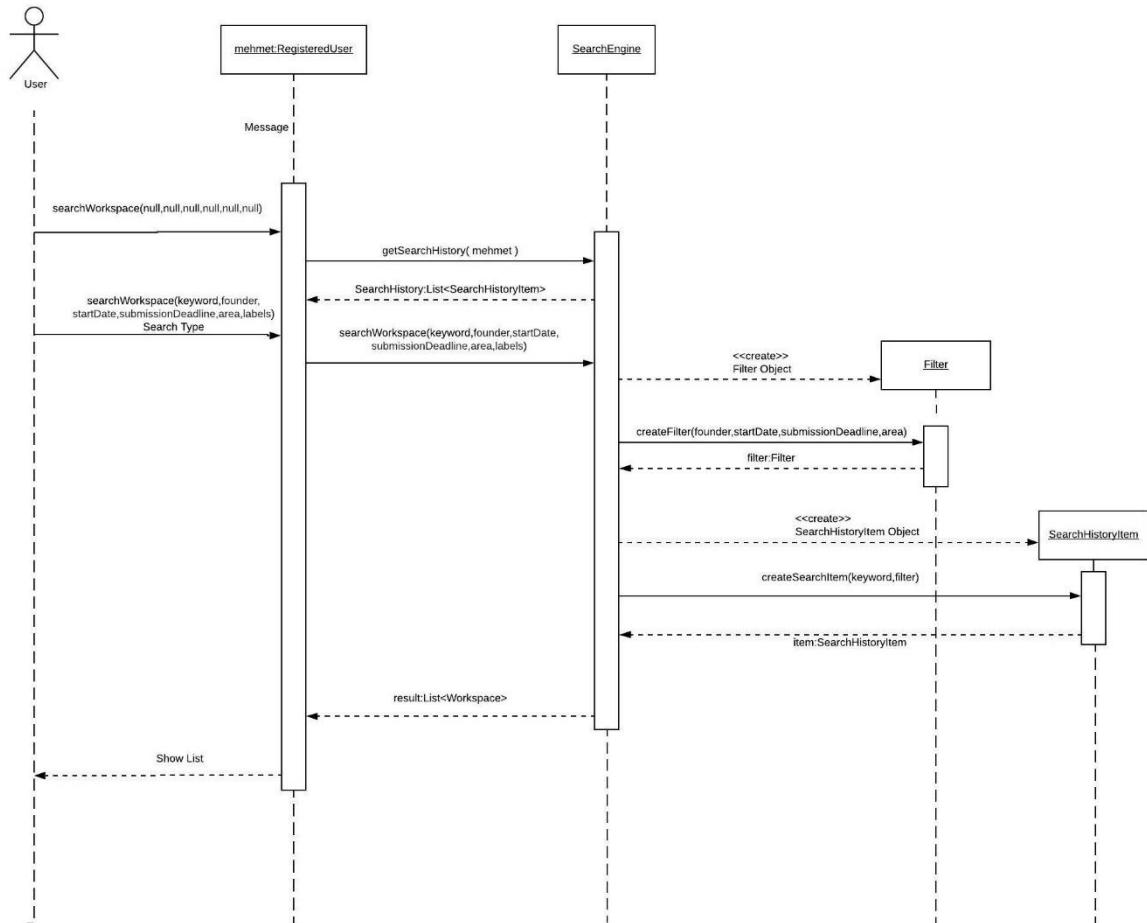
Link Google Scholar Account



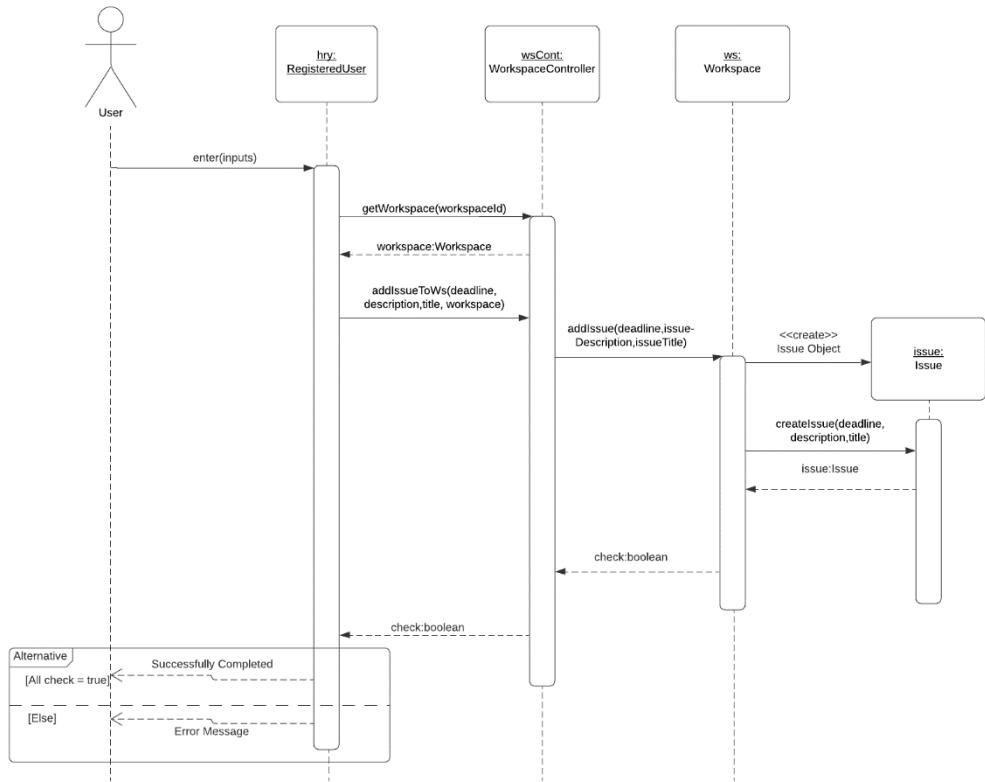
Comment/Rate Profile



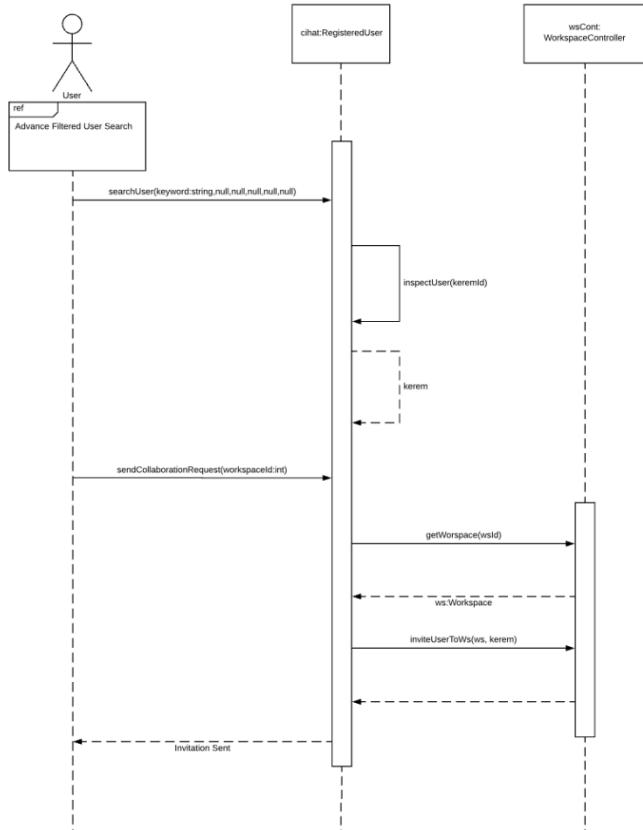
Advanced Filtered Search



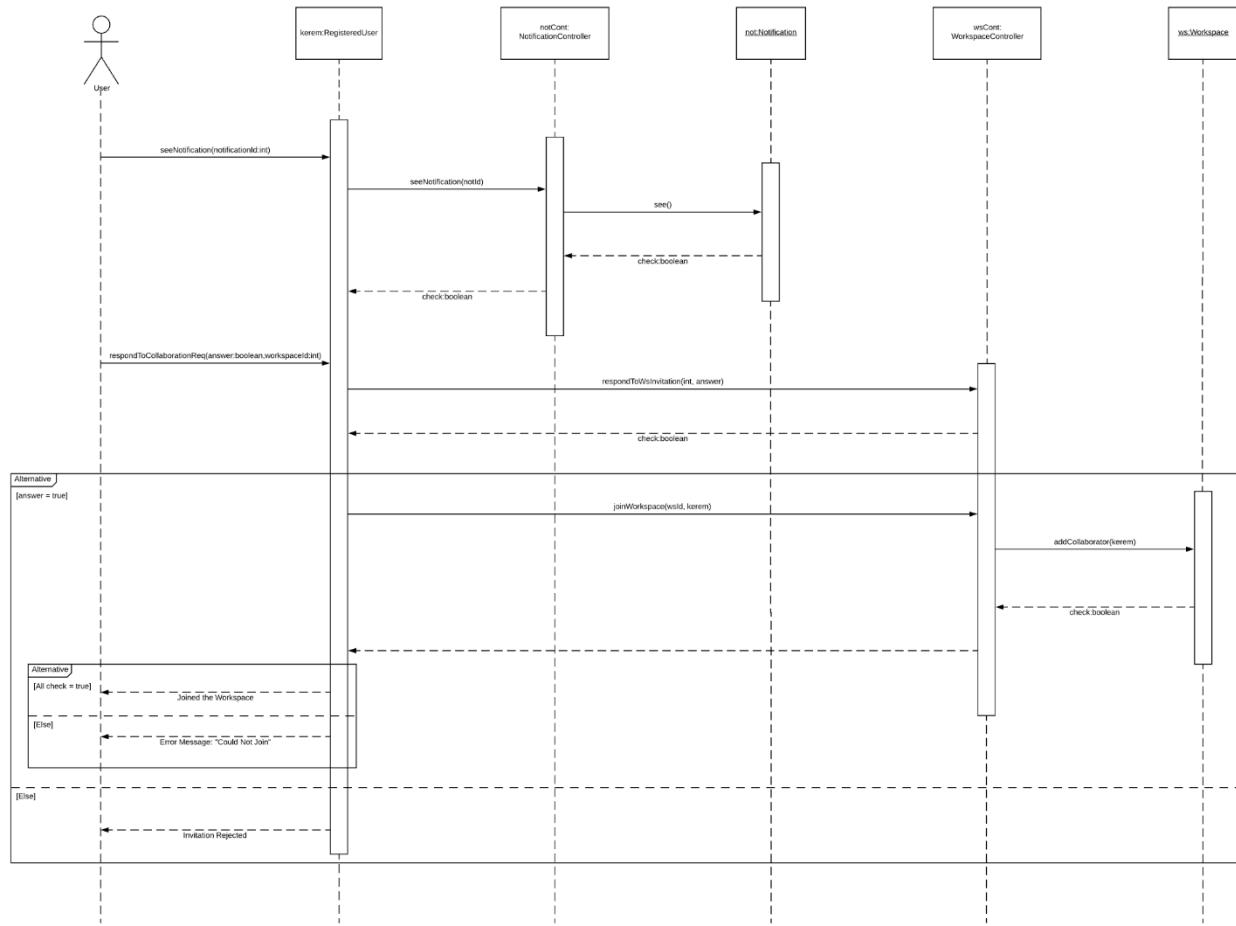
Creating Issue



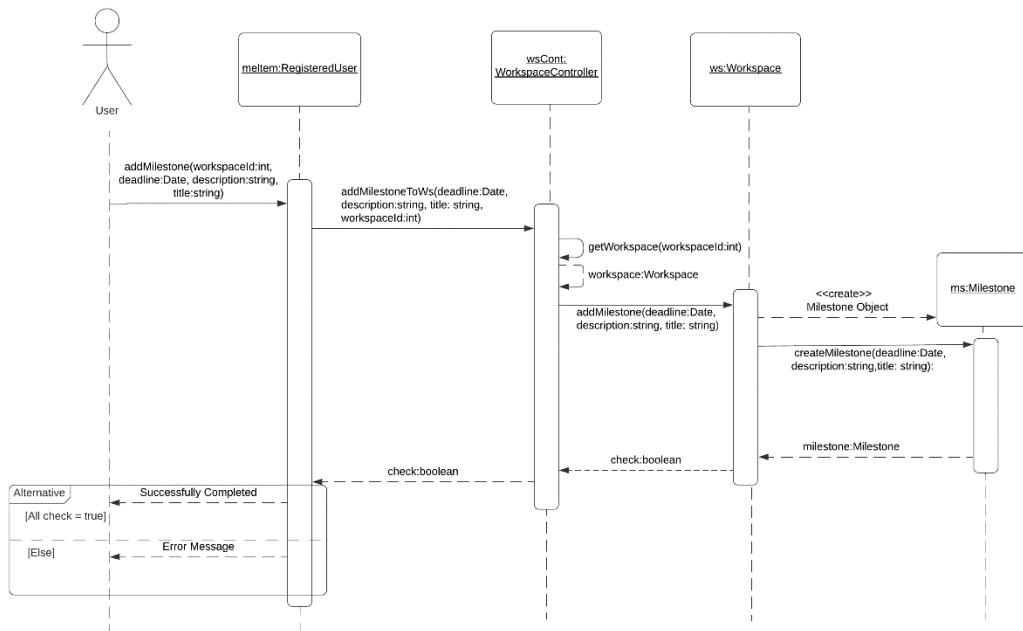
Inviting a User to Workspace



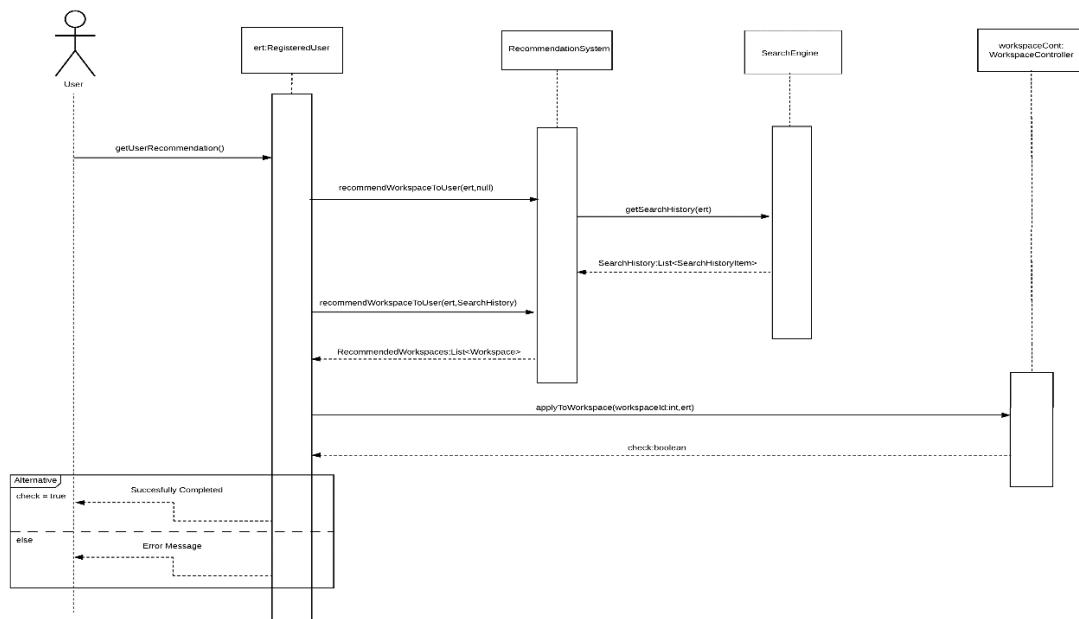
Responding to an Invitation



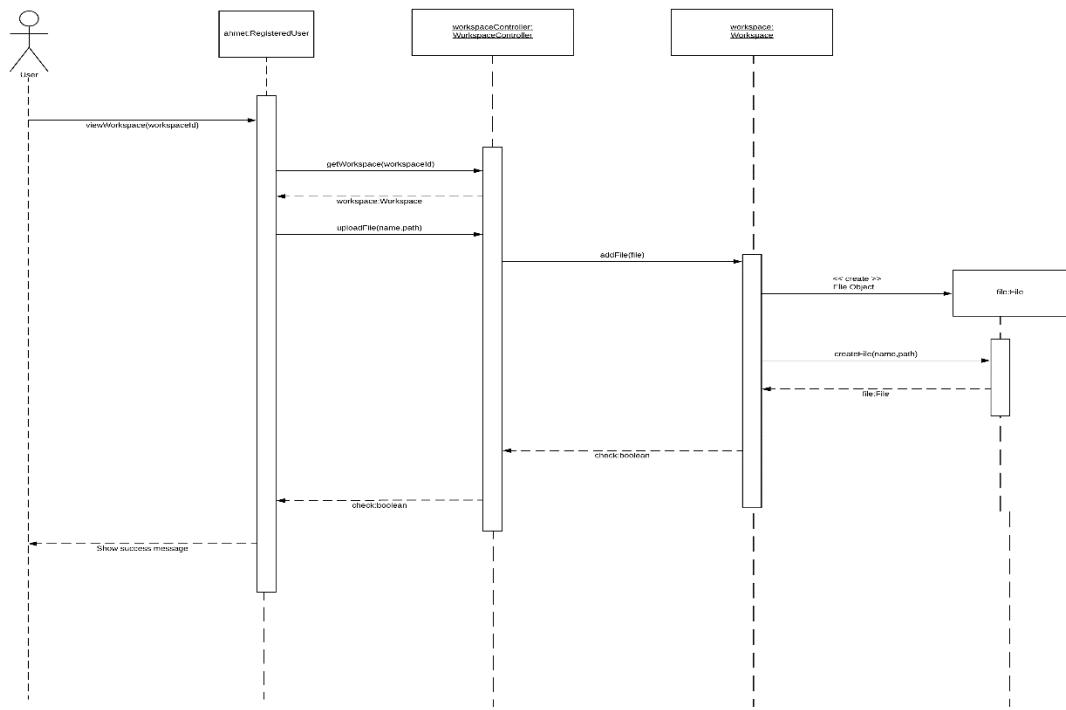
Milestone Creation



Send Collaboration Request



Adding Files



MANUALS

USER MANUAL

WEB

Platon is an academic collaboration platform that unites everyone that has interest in being part of a research project. It is prepared by 7th group for CMPE 352/451 course, Bogazici University students.

For Guests

The screenshot shows the Platon homepage with a dark background. At the top, there is a navigation bar with the word "PLATON" on the left, a search bar in the center containing "Search...", and "LOGIN" and "REGISTER" buttons on the right. Below the navigation bar, the title "Welcome to Platon" is centered. A subtitle below it reads "Discover research and connect with your scientific community." On the left side, there is a section titled "Trending Workspaces" which lists five workspace entries. On the right side, there is a section titled "Upcoming Events" which lists four events. At the bottom of the page, there is a footer with a disclaimer: "Disclaimer: Data is provided by [CERAPI](#)".

Trending Workspaces
Spacing In Basketball Burak Omur - Erfugul Bulbul - Can Bolukbas - Oyku Yilmaz
Forecasting Natural Disasters with Neural Networks Ahmet Dadak - Maxim Telegako
TREC-COVID Challenge Burak Omur - Hizal Demir - Can Bolukbas - Umutcan Uvut - canos bolukbasos - B Y
Efficient Use of Space in Insect Colonies Burhan Akkus
revised decision trees canos bolukbasos

Upcoming Events
ICHSMT21: International Congress on Health Sciences and Medical Technologies Start Date: Jun 15, 2021 - Jun 17, 2021
National Conference on applied mathematics and Mathematics Modeling Start Date: Mar 30, 2021 - Mar 31, 2021
ACM SIGPLAN International Symposium on Memory Management Start Date: Jun 22, 2021 - Jun 22, 2021
World Congress on Special Needs Education Start Date: Nov 22, 2021 - Nov 24, 2021
IEEE International Conference on Multimedia Big Data 2021 Start Date: Nov 15, 2021 - Nov 17, 2021

If you do not have an account yet, you can still use some functionalities of Platon. You can search for users, upcoming events, and even workspaces that are currently being worked on or published. You can see the trending projects.

The screenshot shows the Platon homepage. At the top, there is a navigation bar with the word "PLATON" on the left, a search bar in the center, and "LOGIN" and "REGISTER" buttons on the right. Below the navigation bar, the main heading "Welcome to Platon" is displayed, followed by the subtext "Discover research and connect with your scientific community." On the left side, there is a section titled "Trending Workspaces" which lists five workspace entries. On the right side, there is a section titled "Upcoming Events" which lists several events with their names, start dates, and end dates. A disclaimer at the bottom right states "Disclaimer: Data is provided by CTP&R".

Trending Workspaces
Spacing In Basketball Burak Ömür - Ertegun Bulbul - Can Bolukbas - Cyku Yilmaz
Forecasting Natural Disasters with Neural Networks Ahmet Davlek - Maxim Trigalko
TREC-COVID Challenge Burak Ömür - Hul Derrin - Can Bolukbas - Umutcan Uvut - canos bolukbasos - B Y
Efficient Use of Space in Insect Colonies Burhan Akkus
revised decision trees canos bolukbasos

Upcoming Events
ICHSMT21: International Congress on Health Sciences and Medical Technologies Start Date: Jun 15, 2021 - Jun 17, 2021
National Conference on applied mathematics and Mathematics Modeling Start Date: Mar 30, 2021 - Mar 31, 2021
ACM SIGPLAN International Symposium on Memory Management Start Date: Jun 22, 2021 - Jun 22, 2021
World Congress on Special Needs Education Start Date: Nov 22, 2021 - Nov 24, 2021
IEEE International Conference on Multimedia Big Data 2021 Start Date: Nov 15, 2021 - Nov 17, 2021

If you want to be a part of Platon, you may register by providing your name, surname, email and your job. To go to the register page, click register on the right top.

The screenshot shows the "Register" page of the Platon website. At the top, there is a navigation bar with the word "PLATON" on the left, a search bar in the center, and "LOGIN" and "REGISTER" buttons on the right. Below the navigation bar, the page title "Register" is displayed with a user icon. The form consists of several input fields: "First Name *", "Last Name *", "Email Address *", "Password *", and "Job *". Below the form is a checkbox labeled "I accept the terms and conditions". At the bottom of the page, there is a "REGISTER" button and a link to "Terms and Conditions".

Further, you will be able to link your Google Scholar and ResearchGate accounts to fetch your projects. You have to give an email address in use, since a verification code will be sent to it. You have to verify your account to log in. You have to accept Terms&Conditions to register.



Verify Account

[VERIFY MY ACCOUNT](#)[!\[\]\(90e0b73e8bf809636c3a82a9b5718967_img.jpg\) GO TO MAIN PAGE](#)

After registering, there will be an email sent to you in order to verify your account. By clicking the link provided for you, you can verify your account.

The screenshot shows the Platon homepage. At the top, there is a navigation bar with the Platon logo, a search bar, and login/register buttons. Below the navigation bar, the main heading "Welcome to Platon" is displayed, followed by the subtext "Discover research and connect with your scientific community." On the left side, there is a section titled "Trending Workspaces" which lists five workspace cards. On the right side, there is a section titled "Upcoming Events" which lists six event cards. A navigation bar at the bottom of the page shows page 1 of 104.

Trending Workspaces

- Spading in Basketball
Burak Ömür - Ertegun Bulbul - Can Bolukbas - Ceylu Yilmaz
- Forecasting Natural Disasters with Neural Networks
Ahmet Davlek - Maxim Trigalov
- TREC-COVID Challenge
Burak Ömür - Hul Derrin - Can Bolukbas - Umutcan Uvut - canos bolukbasos - B Y
- Efficient Use of Space in Insect Colonies
Burhan Akkus
- revised decision trees
canos bolukbasos

Upcoming Events

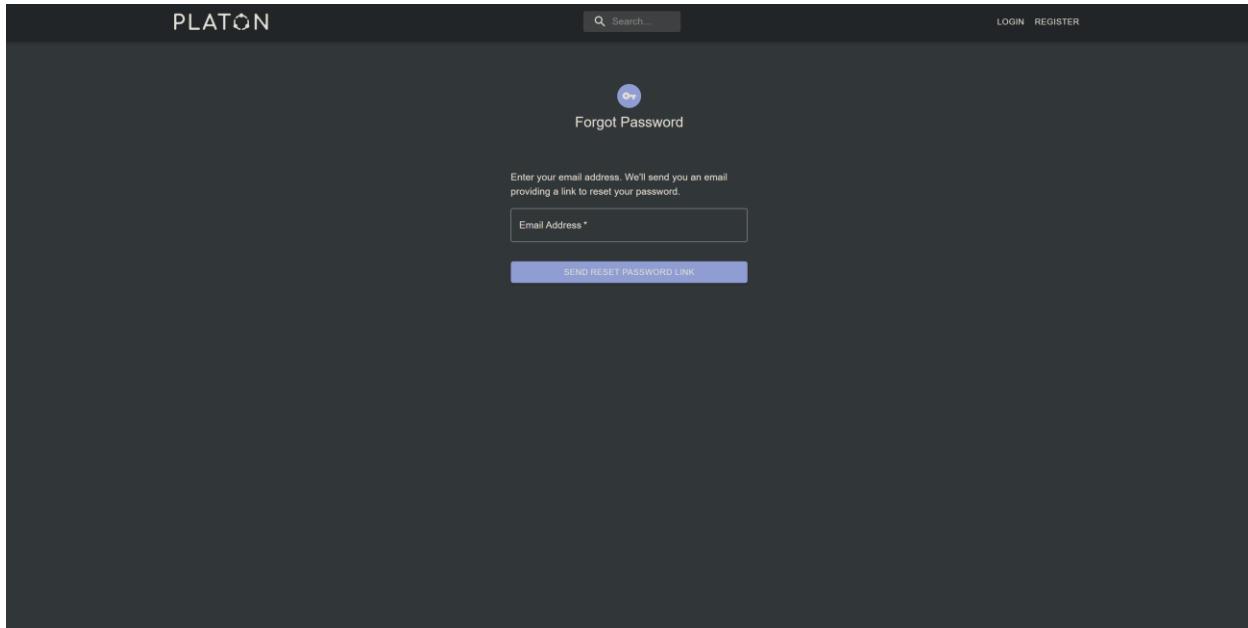
- ICHSMT21: International Congress on Health Sciences and Medical Technologies
Start Date: Jun 15, 2021 - Jun 17, 2021
- National Conference on applied mathematics and Mathematics Modeling
Start Date: Mar 30, 2021 - Mar 31, 2021
- ACM SIGPLAN International Symposium on Memory Management
Start Date: Jun 22, 2021 - Jun 22, 2021
- World Congress on Special Needs Education
Start Date: Nov 22, 2021 - Nov 24, 2021
- IEEE International Conference on Multimedia Big Data 2021
Start Date: Nov 15, 2021 - Nov 17, 2021

< 1 2 3 4 5 ... 104 >

Disclaimer: Data is provided by CTPATH

The screenshot shows the Platon login page. It features a "Login" button with a key icon, two input fields for "Email Address" and "Password", and a "SIGN IN" button. Below the password field is a link for "Forgot password?".

Platon does not require a username for the users. You may log in with providing your email address and password. To log in, click on the LOGIN button on the right top.



If you ever forget your password, you can reset it by clicking on a link that will be sent to your e-mail address.

A screenshot of the PLATON application's home page. It features three main sections: 'Trending Workspaces', 'What's Happening', and 'Upcoming Events'.

- Trending Workspaces:** Shows three workspace cards:
 - 'Predicting Earthquake with DL' by Gwen Stacy - Christophe Daussy
 - 'Medical Information System' by Ahmet Dadak - Burak Ömür - Umutcan Uvut - Oykù Yılmaz - Halil Umut Özdemir - Burhan Can Akkus - Ertugrul Bulbul
 - 'Discovering Bias in Datasets' by Umutcan Uvut - Oykù Yılmaz - Halil Umut Özdemir - Christophe DaussyA pagination bar at the bottom shows pages 1, 2, 3, 4, >.
- What's Happening:** Shows activity feed items:
 - Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy created a new folder named 'Data Analysis' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy created a new folder named 'Initial Model Analysis' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy deleted the file named 'IMG-20190609-WA0001.jpg' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy deleted the file named 'Black Heart Emoji On Apple iOS 11.3.jpg' in the workspace 'Predicting Earthquake with DL'.
 - can bolukbas commented and rated Halil Umut Özdemir.
- Upcoming Events:** Shows a single event card:
 - ACM Conference on Security and Privacy in Wireless and Mobile NetworksWith details: Start Date: Jun 28, 2021 - Jul 1, 2021. A disclaimer at the bottom states: 'Disclaimer: Data is provided by CFP API.'
- Upcoming Deadlines:** Shows three deadline cards:
 - 'Predicting Earthquake with DL' Milestone Deadline: Examine Dataset 11 Dec 2020 02:59
 - 'Predicting Earthquake with DL' Milestone Deadline: Train a Neural Network 24 Dec 2020 02:59
 - 'Predicting Earthquake with DL' Milestone Deadline: Analysis of Earthquake Predictions 1 Jan 2021 02:59A pagination bar at the bottom shows pages 1, 2, >.

You will be directed to the home page.

Your Profile

The screenshots illustrate the PLATON platform's user interface, designed for researchers and professionals.

Top Screenshot (Homepage):

- Trending Workspaces:** Shows three workspace cards:
 - Predicting Earthquake with DL by Gwen Stacy - Christophe Daussy
 - Medical Information System by Ahmed Dadak - Burak Ömür - Umutcan Uvut - Oykù Yılmaz - Hali Umut Özdemir - Can Akkus - Ertugrul Bülbül
 - Discovering Bias in Datasets by Umutcan Uvut - Oykù Yılmaz - Hali Umut Özdemir - Christophe Daussy
- What's Happening?** A feed of recent activity:
 - Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy created a new folder named 'Data Analysis' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy created a new folder named 'Initial Model Analysis' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy deleted the file named 'IMG-20190609-WA0001.jpg' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy deleted the file named 'Black Heart Emoji On Apple Ios 11.3.jpg' in the workspace 'Predicting Earthquake with DL'.
 - can bolukbas commented and rated Hali Umut Özdemir
- Upcoming Events:** A list of events:
 - ACM Conference on Security and Privacy in Wireless and Mobile Networks (Start Date: Jun 28, 2021 - Jul 1, 2021)

Bottom Screenshot (Profile Page for Gwen Stacy):

- User Profile:** Gwen Stacy, Graduate Student - Chicago State University - CS. Specialties: Python, C, Deep Learning, Conv., MATLAB, AI.
- Profile Metrics:** 8 Followers, 7 Following. Edit Profile button.
- Profile Rating:** ★★★★☆
- Profile Sections:** RESEARCHS (active), FOLLOWERS, FOLLOWING, COMMENTS, REQUESTS, RECOMMENDATION.
- CREATE RESEARCH:** Button to start a new research entry.
- Research Entry (Title: Hierarchical Mixtures Of Generators For Adversarial Learning):** Description: No description is provided. Year: 2019. UPDATE RESEARCH and Delete buttons.

To go to your profile, click on the profile icon on the right top.

You may personalize your profile by several ways:

The screenshots illustrate the PLATON profile editing interface. The top part shows the user's basic profile information, including their name, job title, and a list of skills they possess. The bottom part shows the detailed edit profile form, where users can update their personal information, upload a profile photo, and link their academic profiles. The 'Edit Skills' section allows users to manage the list of skills displayed on their profile.

If you are not satisfied with your current information, you may click "Edit Profile" button in your profile page. Here, you may add some of your skills, change your current information, upload a profile photo, and link your Google Scholar and ResearchGate accounts. You can change the privacy of your profile (see Engaging with Other Profiles). If you link your account, the projects you registered there will be automatically uploaded to your profile. You may add yourself an existing skill, or you may create a skill. You can choose from the list for existing skills, or just type the skill name you want to add and then click the add button.

PLATON

Profile

Gwen Stacy
Graduate Student - Chicago State University - CS
Python C Deep Learning Cnn MATLAB AI

8 Followers 7 Following

Edit Profile ★★★★☆

RESEARCHES FOLLOWERS FOLLOWING COMMENTS REQUESTS RECOMMENDATION

CREATE RESEARCH

Title
Hierarchical Mixtures Of Generators For Adversarial Learning

Description
No description is provided.

Year
2019

UPDATE RESEARCH Delete

PLATON

Profile

Gwen Stacy
Graduate Student - Chicago State University - CS
Python C Deep Learning Cnn MATLAB AI

8 Followers 7 Following

Edit Profile ★★★★☆

RESEARCHES FOLLOWERS FOLLOWING COMMENTS REQUESTS RECOMMENDATION

Add New Research

Title
Hierarchical M

Description
Hierarchical M

Year
2019

CREATE

UPDATE RESEARCH Delete

Title
Deepsym: Deep Symbol Generation And Rule Learning From Unsupervised Continuous Robot Interaction For Planning

Description

You may also click the "Create Research" button on your profile to add your past research. You need to provide your project's title, a description of it, and also a date. Date and title are mandatory.

PLATON

Profile

Gwen Stacy
Graduate Student - Chicago State University - CS
Python C Deep Learning Cnn MATLAB AI

8 Followers 7 Following

Edit Profile ★★★★☆

RESEARCHES FOLLOWERS FOLLOWING COMMENTS REQUESTS RECOMMENDATION

CREATE RESEARCH

Title
Hierarchical Mixtures Of Generators For Adversarial Learning

Description
No description is provided.

Year
2019

UPDATE RESEARCH Delete

Title

PLATON

Profile

Gwen Stacy
Graduate Student - Chicago State University - CS
Python C Deep Learning Cnn MATLAB AI

8 Followers 7 Following

Edit Profile

RESEARCHES FOLLOWERS FOLLOWING COMMENTS REQUESTS RECOMMENDATION

Edit Research

Title Hierarchical Mixtures Of G

Description

Year 2019

UPDATE

Year
2019

UPDATE RESEARCH Delete

Title
Deepsym: Deep Symbol Generation And Rule Learning From Unsupervised Continuous Robot Interaction For Planning

Description

You can edit your research information later on by clicking "Update Research" button.

PLATON

Profile

Gwen Stacy
Graduate Student - Chicago State University - CS
Python C Deep Learning Cnn MATLAB AI

8 Followers 7 Following

Edit Profile

★★★★★

RESEARCHES FOLLOWERS FOLLOWING COMMENTS REQUESTS RECOMMENDATION

CREATE RESEARCH

Title
Hierarchical Mixtures Of Generators For Adversarial Learning

Description
No description is provided.

Year
2019

UPDATE RESEARCH

Delete

Title

You can also delete it by clicking the delete button, if you change your mind.

Engaging with Other Profiles

There are two types of profiles in Platon: public and private. If you choose the privacy of your account as public, other profiles can see your name, surname, job, institution, skills, followers, following, and past research information. Registered users will be able to follow you without sending a follow request first.

If you want a private profile, other users and guests can only view your name, surname, profile photo and your job. To view your other personal information, they will have to send you a follow request. You will have the option to accept and reject their request.

If you accept, they will view your information as if you had a public account.

PLATON

Profile

Gwen Stacy
Graduate Student - Chicago State University - CS
Python C Deep Learning Cnn MATLAB AI

8 Followers 7 Following

Edit Profile ★★★★★

RESEARCHES FOLLOWERS FOLLOWING COMMENTS REQUESTS RECOMMENDATION

CREATE RESEARCH

Title
Hierarchical Mixtures Of Generators For Adversarial Learning

Description
No description is provided.

Year
2019

UPDATE RESEARCH Delete

Title

PLATON

Profile

Gwen Stacy
Graduate Student - Chicago State University - CS
Python C Deep Learning Cnn MATLAB AI

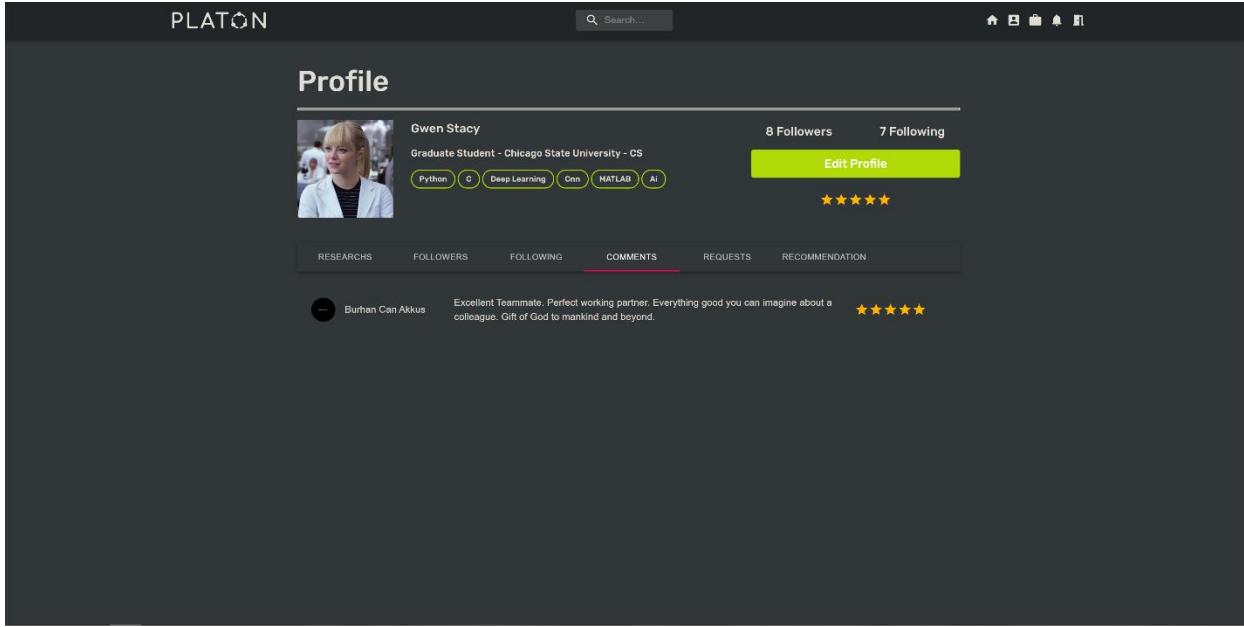
8 Followers 7 Following

Edit Profile ★★★★★

RESEARCHES FOLLOWERS FOLLOWING COMMENTS REQUESTS RECOMMENDATION

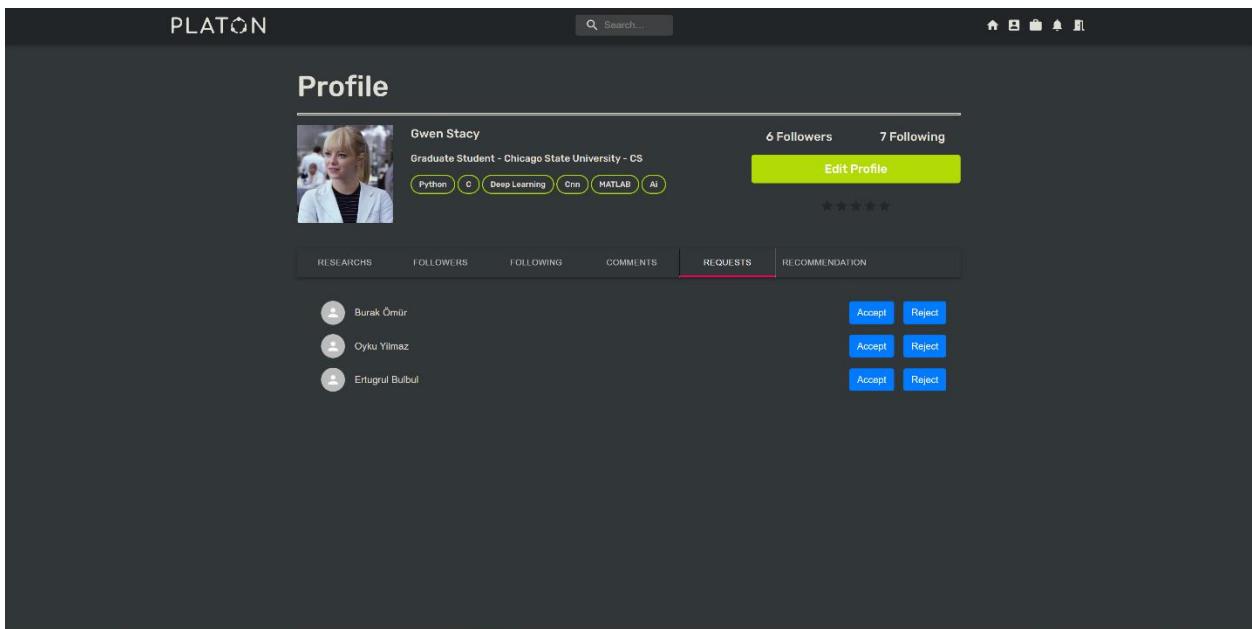
- Ahmet Dadak
- Burak Ömür
- cen bolukbes
- Sene Ergisi
- Burhan Can Akkus
- Ertugrul Bulbul
- Christophe Daussy
- Aiperen Divriklioglu

You can see your followers and following in your profile under the tabs named "FOLLOWERS", "FOLLOWING", respectively.



The screenshot shows a user profile page on the PLATON platform. At the top, there's a search bar and several icons. Below the header, the title "Profile" is displayed above a user's profile picture of Gwen Stacy. Her name and title ("Graduate Student - Chicago State University - CS") are listed, along with her follower count (8 Followers) and the number of users she follows (7 Following). A green "Edit Profile" button is visible. Below this, there are tags for programming languages and technologies: Python, C, Deep Learning, Conv., MATLAB, and AI. A five-star rating is shown. A comment from a user named Burhan Can Akkus is displayed, followed by another five-star rating. At the bottom of the profile section, there are tabs for RESEARCHES, FOLLOWERS, FOLLOWING, COMMENTS, REQUESTS, and RECOMMENDATION.

Other users you have collaborated before may leave comments and rate you. You cannot alter these comments, or delete them. To see your comments, go to the comments section on your profile.



This screenshot shows the same user profile as the previous one, but the REQUESTS tab is now active. It displays three follow request notifications from users: Burak Ömür, Oyku Yilmaz, and Ertugrul Bulbul. Each notification includes a small profile picture, the user's name, and two buttons: "Accept" and "Reject".

If you have a private account, users cannot directly follow you, you need to accept their requests first. To answer the follow requests, go to the REQUEST section.

The screenshot shows the PLATON profile page for a user named Gwen Stacy. At the top, there's a search bar and a navigation menu with icons for home, search, and notifications. Below the header, the word "Profile" is displayed in a large, bold font. To the left is a profile picture of Gwen Stacy, a young woman with blonde hair. To the right of the picture, her name "Gwen Stacy" is shown, followed by her title "Graduate Student - Chicago State University - CS". Further to the right are two statistics: "8 Followers" and "7 Following". Below these stats is a green button labeled "Edit Profile". Underneath the stats, there are five small yellow star icons. At the bottom of the profile section, there are tabs for "RESEARCHES", "FOLLOWERS", "FOLLOWING", "COMMENTS", "REQUESTS", and "RECOMMENDATION". The "RECOMMENDATION" tab is currently selected and highlighted in red. Below the tabs, there's a list of recommended users, each with a small profile icon and their names: Ahmet Eker, Berkay Karaman, Sene Ergisi, Burhan Can Akkus, and Cagri Ciftci.

Platon offers a recommendation system. You can see your recommendations for following users on the recommendation tab. The suggestions will be according to the match rate of your and other users' skills.

Workspaces

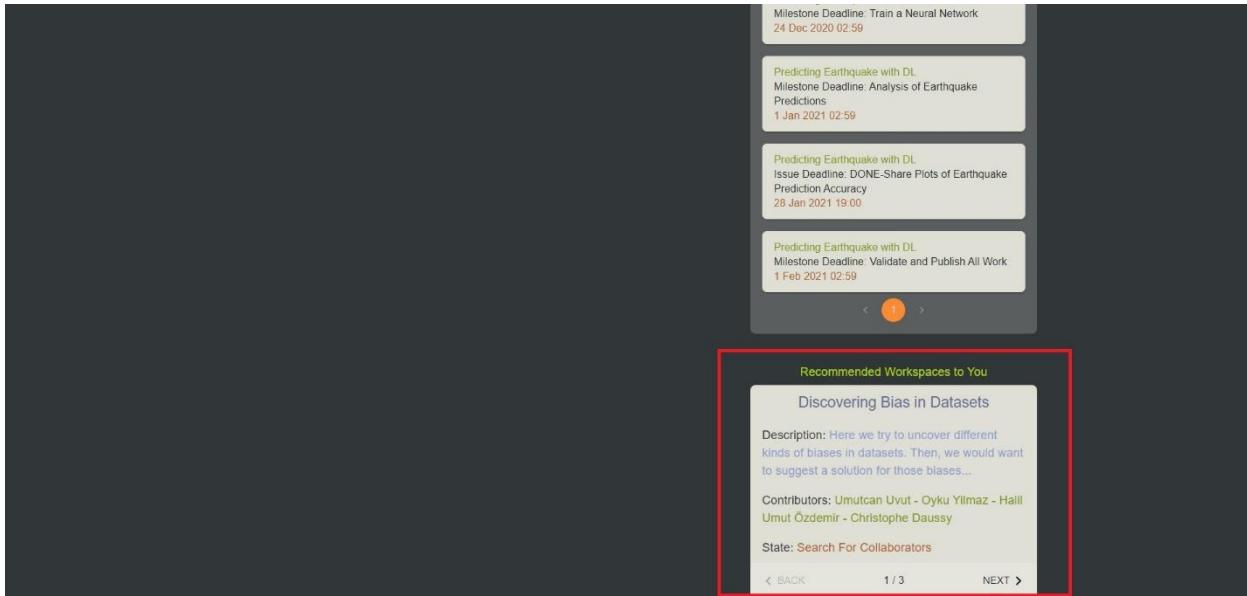
The screenshot shows the PLATON platform interface. At the top, there is a search bar and several icons for navigation and notifications. Below the header, there are three main sections: "Trending Workspaces", "What's Happening?", and "Upcoming Events".

- Trending Workspaces:** Displays three workspace cards:
 - Predicting Earthquake with DL (by Gwen Stacy - Christophe Daussy)
 - Medical Information System (by Ahmet Dadak - Burak Ömür - Umutcan Uvut - Oykù Yılmaz - Halil Ümit Özdemir - Burhan Can Akkus - Ertrugrul Bulbul)
 - Discovering Bias in Datasets (by Umutcan Uvut - Oykù Yılmaz - Halil Ümit Özdemir - Christophe Daussy)A pagination control with pages 1, 2, 3, ..., 231 is shown at the bottom.
- What's Happening? :** Shows activity logs for the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy uploaded a new file named 'paper.pdf'.
 - Christophe Daussy created a new folder named 'Data Analysis'.
 - Christophe Daussy created a new folder named 'Initial Model Analysis'.
 - Christophe Daussy deleted a file named 'IMG-20190609-WA0001.jpg'.
 - Christophe Daussy deleted a file named 'Black Heart Emoji On Apple Ios 11 3.jpg'.
 - can bolukbas commented and rated Halil Umut Özdemir.
- Upcoming Events:** Lists an upcoming conference.
 - ACM Conference on Security and Privacy in Wireless and Mobile Networks (Start Date: Jun 28, 2021 - Jul 1, 2021)A disclaimer at the bottom states: "Disclaimer: Data is provided by CFP API."

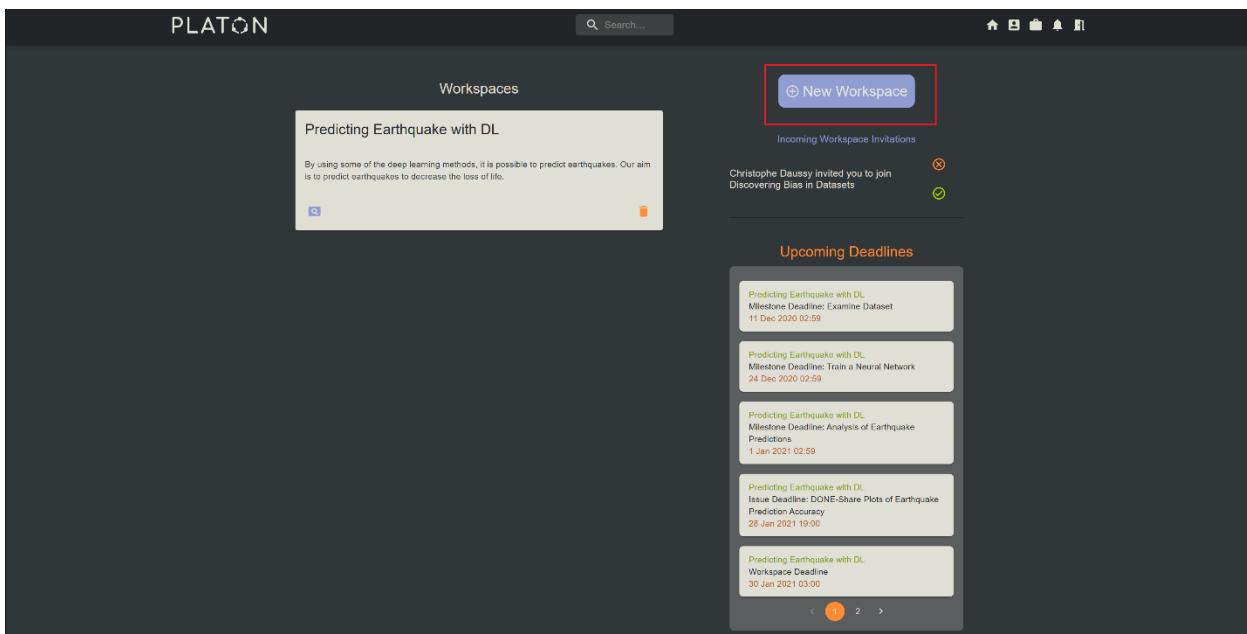
This screenshot shows a detailed view of a workspace titled "Predicting Earthquake with DL".

- Workspace Overview:** Includes a description: "By using some of the deep learning methods, it is possible to predict earthquakes. Our aim is to predict earthquakes to decrease the loss of life.", a "New Workspace" button, and an "Incoming Workspace Invitations" section where Christophe Daussy has invited can bolukbas to join.
- Upcoming Deadlines:** Lists three milestones:
 - Predicting Earthquake with DL, Milestone Deadline: Examine Dataset, 11 Dec 2020 02:59
 - Predicting Earthquake with DL, Milestone Deadline: Train a Neural Network, 24 Dec 2020 02:59
 - Predicting Earthquake with DL, Milestone Deadline: Analysis of Earthquake Predictions, 1 Jan 2021 02:59A pagination control with pages 1, 2, ... is shown at the bottom.

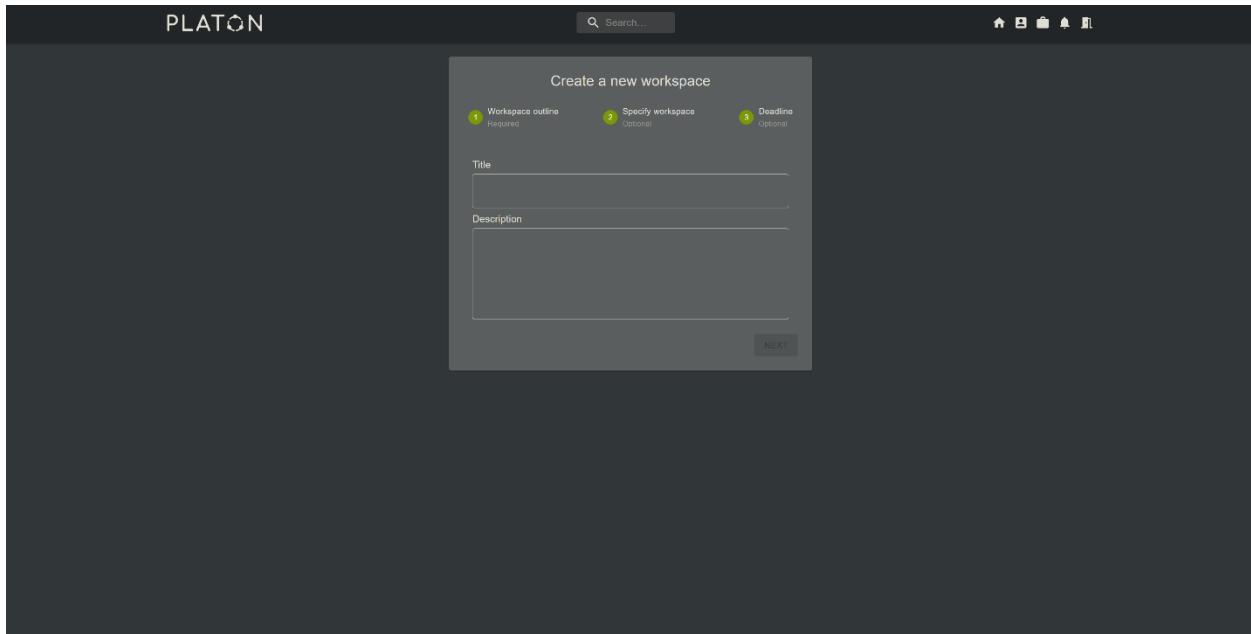
Workspaces are storage areas for a group of users to develop their project/paper proposal collaboratively. To list your workspaces, click on the workspace icon on the top right.



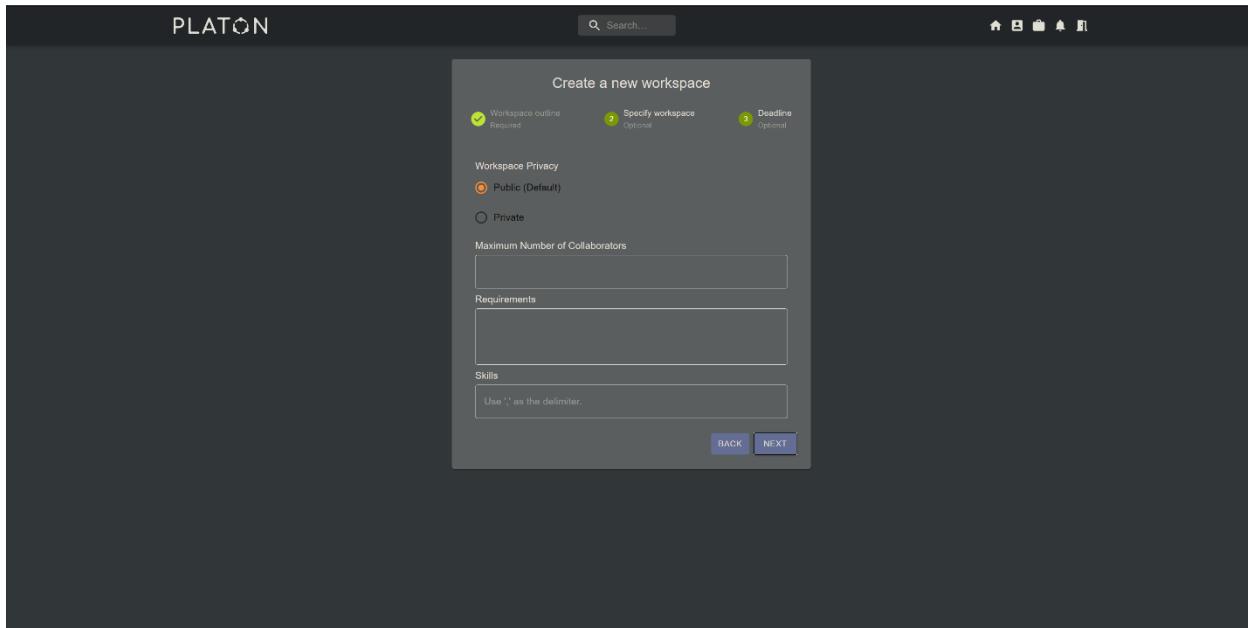
Under the personal reminder section, there is a workspace recommendation section. You can see the workspaces recommended for you.



If you do not have a workspace yet, you can create a workspace by clicking the "NEW WORKSPACE" button.



To create a workspace, you must provide a title and a description associated with the workspace. After clicking next, you may provide other information.



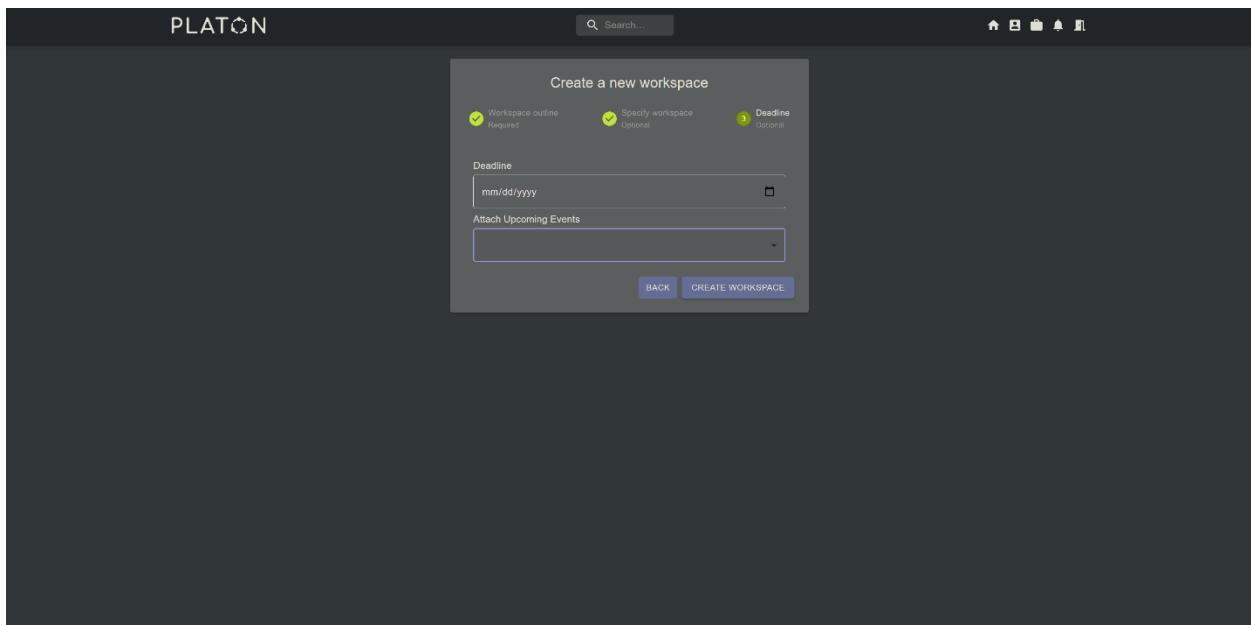
The workspace is created as public by default. Public workspace means any user or guest may view the details of your workspace, files, issues, milestones, and the contributors in your workspace. They may search for it from the search bar, and your workspace may be seen in the trending projects, if it gets much attention. After viewing, a user may send a collaboration request to join your workspace.

If you want your workspace to be invisible, you should make it private. Private workspaces does not show up in any search or in the trending projects section. Only the collaborators may view it.

You can provide the maximum number of collaborators in your workspace. The default number is 10 if you decide to leave it empty.

You may write the requirements to join the workspace under the Requirements section.

You may add the required skills in your workspace. Skills will act as labels to your workspace. If you set your workspace to be public, other users may use those skills to narrow down their search. For example, if your workspace has a skill named "Python", then if the user filters their search by giving python as a parameter, your workspace will be visible to their search result.



You may provide a deadline to your workspace.

You may link one or several upcoming events to your workspace. That will mean the project will also be a part of the specified upcoming event/s.

The screenshot shows the PLATON Workspaces interface. At the top, there's a search bar and a navigation bar with icons for home, file, settings, and notifications. Below the header, the title "Workspaces" is displayed. A workspace card for "Predicting Earthquake with DL" is shown, featuring a thumbnail with a red square icon, the workspace title, a brief description, and a status indicator. To the right of the workspace card, there's a button labeled "+ New Workspace". Below the workspace card, there's a section for "Incoming Workspace Invitations" with a message from Christophe Daussy. Further down, there's a section for "Upcoming Deadlines" listing four tasks related to the workspace, each with a deadline and status. At the bottom of the workspace card area, there are navigation arrows.

To inspect a workspace, click on the inspect button.

The screenshot shows the "View Workspace" interface for the "Predicting Earthquake with DL" workspace. The top navigation bar includes "View Workspace", "Edit workspace", and tabs for "DETAILS", "FILES", "ISSUES", and "MILESTONES". The "DETAILS" tab is active. The workspace details include the title, a brief description, workspace privacy set to "Public", and a maximum collaborator limit of "2". Under "Requirements", there are two items: "Knowledge of Deep Learning - Knowledge of Data Science" and "Knowledge of Signal Processing". Under "Tags", there are several categories like Python, Java, C++, Machine Learning, Deep Learning, and Artificial Intelligence. The "Deadline" is listed as "30.01.2021". Under "Attached Learning Events", there are links to "ML4H 2021", "ASB 2021", and "DMA 2021". The "Collaborators" section lists "Gwen Stacy" and "Christophe Daussy", with a "EDIT WORKSPACE" button. The "Incoming Requests" section shows a request from "Oyku Yilmaz". The "State" section indicates the workspace is "Search for Collaborators" (green), "Ongoing" (green), and "Published" (orange). There are also "EDIT WORKSPACE" and "SEARCH" buttons at the bottom.

Here you will see the details of your workspace.

PLATON

[View Workspace](#) [Back to workspace](#)

Predicting Earthquake with DL

By using some of the deep learning methods, it is possible to predict earthquakes. Our aim is to predict earthquakes to decrease the loss of life.

Workspace Privacy: Public Maximum Collaborator: 2

Requirements: Knowledge of Deep Learning, Knowledge of Data Science, Knowledge of Signal Processing

Tags: Python, Deep Learning, Machine Learning, MATLAB, Artificial Intelligence

Deadline: 30.01.2021

Attached Upcoming Events: aIDM 2021, MIWAI 2021, ASDI 2021, DMA 2021

Contributors: Genn Stacy, Christophe Duvay [Edit workspace](#)

Incoming Requests: Oyku Yilmaz [Edit workspace](#)

State: Published

Search For Collaborators: [Edit workspace](#)

On going: [Edit workspace](#)

Published: [Edit workspace](#)

EDIT WORKSPACE

PLATON

[Search...](#) [Back to workspace page](#)

Edit Workspace

Title: Predicting Earthquake with DL

Description: By using some of the deep learning methods, it is possible to predict earthquakes. Our aim is to predict earthquakes to decrease the loss of life.

State: Published

Workspace Privacy: Public (Default) Private

Maximum Number of Collaborators: 2

Requirements: Knowledge of Deep Learning, Knowledge of Data Science, Knowledge of Signal Processing

Skills: Python, Deep Learning, Machine Learning, MATLAB, Artificial Intelligence

Deadline: 01/30/2021

Attach Upcoming Events: aIDM 2021, DMA 2021, MIWAI 2021, ASDI 2021

EDIT

To edit your workspace you need to click the **EDIT WORKSPACE** button.

The screenshot shows a workspace titled "Predicting Earthquake with DL". The workspace details include:

- Workspace Privacy:** Public
- Maximum Collaborator:** 2
- Requirements:** Knowledge of Deep Learning, Knowledge of Data Science, Knowledge of Signal Processing
- Tags:** Python, Deep Learning, Machine Learning, AI, ML, Artificial Intelligence
- Deadline:** 30.01.2021
- Attached Upcoming Events:** IJCAI 2021, NeurIPS 2021, AAAI 2021, ICML 2021
- Collaborators:** Cesar Ramy, Christophe Dauby
- Incoming Requests:** Duygu Yilmaz
- State:**
 - Search For Collaborators (green circle)
 - Ongoing (green circle)
 - Published (orange circle)

A workspace has three states:

- 1- Search for Collaborators State: It is the first of the three states of a project. In this state, the founder of the project waits for requests from other Users and sends invitations to Users. A workspace is created with this state by default.
- 2- Ongoing State: It is the second of the three states of a project. This is the development phase of the project. After finding enough collaborators, the project will go to the second state which is the Ongoing State.
- 3- Published State: It is the last stage of a project. After milestones are completed with the request of owners of the project, the project will go to Published State.

You may change the state of the project by editing your workspace.

PLATON

View Workspace

Back to workspaces

DETAILS FILES ISSUES MILESTONES

Predicting Earthquake with DL

By using some of the deep learning methods, it is possible to predict earthquakes. Our aim is to predict earthquakes to decrease the loss of life.

Workspace Privacy: Public Maximum Collaborator: 2

Requirements: - Knowledge of Deep Learning - Knowledge of Data Science - Knowledge of Signal Processing

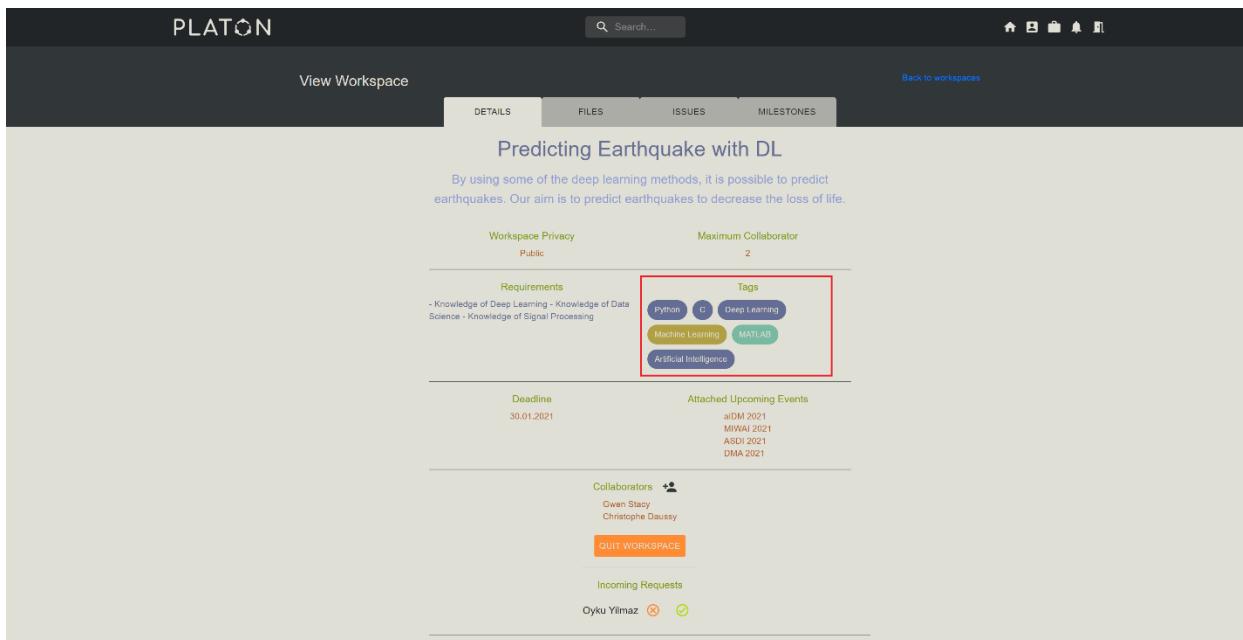
Tags: Python, C, Deep Learning, Machine Learning, MATLAB, Artificial Intelligence

Deadline: 30.01.2021 Attached Upcoming Events: aiDM 2021, MIWAI 2021, ASDI 2021, DMA 2021

Collaborators: Gwen Stacy, Christophe Daussy

QUIT WORKSPACE

Incoming Requests: Oyku Yilmaz



PLATON

View Workspace

Back to workspace

DETAILS FILES ISSUES MILESTONES

Predicting Earthquake with DL

By using some of the deep learning methods, it is possible to predict earthquakes. Our aim is to predict earthquakes to decrease the loss of life.

Workspaces with Python:

- Medical Information System (Owner: Ahmet Dadak)
- Spading in Basketball (Owner: Ertegun Bulbul)
- Discovering Bias in Datasets (Owner: Christophe Daussy)
- Image Denoising (Owner: Ertegun Bulbul)
- TREC COVID (Owner: can bolukbas)

Maximum Collaborator: 2

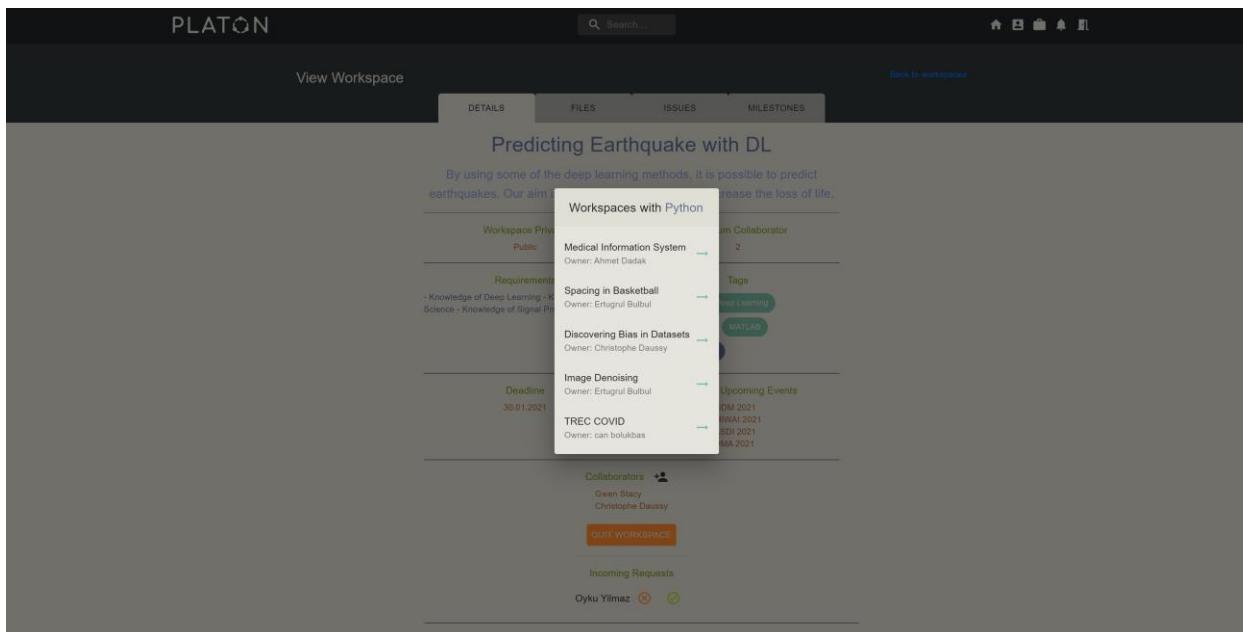
Tags: Deep Learning, MATLAB, Python

Upcoming Events: aiDM 2021, MIWAI 2021, ASDI 2021, DMA 2021

Collaborators: Gwen Stacy, Christophe Daussy

QUIT WORKSPACE

Incoming Requests: Oyku Yilmaz



Tags on the detail section are clickable. By clicking on them you can see the workspaces that having the same tag.

The screenshot shows the PLATON workspace interface for a project titled "Predicting Earthquake with DL". The workspace is set to "Public" and has a maximum collaborator limit of 2. It includes sections for "Requirements" (listing knowledge of Deep Learning, Data Science, Signal Processing) and "Tags" (Python, C, Deep Learning, Machine Learning, MATLAB, Artificial Intelligence). A "Deadline" is listed as 30.01.2021. Attached upcoming events include aiDM 2021, MIWAI 2021, ASDI 2021, and DMA 2021. Collaborators listed are Gwen Stacy and Christophe Daussy.

Platon also has a recommendation system for workspaces. It recommends users to workspaces to collaborate by checking the match rate of the tags of a workspace and the skills of the users.

The screenshot shows the same workspace interface, but a modal window titled "Recommended Users for Collaboration" is open over the main content. This modal lists five recommended users with their profiles and roles: Sena Ergisi (Electrical Engineer @Washington University in St. Louis), Ertugrul Bulbul (Computer Engineer @Bogazici University), Oyku Yilmaz (Student @Bogazici University), Helli Umut Ozdemir (Student @Bogazici University), and Burhan Can Akkus (Pianist). Each user entry includes a small profile picture, their name, title, and the name of their current institution.

When you click on the button, you may see the suggestions.

The screenshot shows the PLATON workspace interface. At the top, there's a dark header bar with the PLATON logo, a search bar, and several icons. Below it, a secondary navigation bar has 'View Workspace' on the left and 'Back to workspaces' on the right. The main content area is titled 'Predicting Earthquake with DL'. It includes a brief description: 'By using some of the deep learning methods, it is possible to predict earthquakes. Our aim is to predict earthquakes to decrease the loss of life.' Below this, there are two sections: 'Workspace Privacy' (set to 'Public') and 'Maximum Collaborator' (set to '2'). The 'Requirements' section lists '- Knowledge of Deep Learning - Knowledge of Data Science - Knowledge of Signal Processing'. The 'Tags' section includes Python, C, Deep Learning, Machine Learning, MATLAB, and Artificial Intelligence. Under 'Deadline', it says '30.01.2021'. In the 'Attached Upcoming Events' section, there are four entries: aiDM 2021, MIWAI 2021, ASDI 2021, and DMA 2021. The 'Collaborators' section shows Gwen Stacy and Christophe Daussey. At the bottom, there's a 'QUIT WORKSPACE' button and an 'Incoming Requests' section which is currently empty.

If your workspace is public, users are able to send it a collaboration request.

You can view your workspace's applications in its details page. You may accept or reject the application.

If you accept, the user will automatically be added as collaborator to your workspace.

If you get accepted, you will automatically be added as collaborator to their workspace.

There is no limitation on the number of applications, invitations, or on the number of your workspaces.

The screenshot shows the 'View Workspace' interface for a workspace named 'Predicting Earthquake with DL'. At the top, there are tabs for 'DETAILS', 'FILES', 'ISSUES', and 'MILESTONES'. Below the tabs, a summary section contains the following information:

- Workspace Privacy:** Public
- Maximum Collaborator:** 2
- Requirements:** - Knowledge of Deep Learning - Knowledge of Data Science - Knowledge of Signal Processing
- Tags:** Python, C, Deep Learning, Machine Learning, MATLAB, Artificial Intelligence
- Deadline:** 30.01.2021
- Attached Upcoming Events:** IJCAI 2021, MWAI 2021, ASDI 2021, DMA 2021
- Collaborators:** Gwen Stacy, Christophe Daussy

A prominent orange button labeled 'QUIT WORKSPACE' is centered at the bottom of the summary section. Below this, under 'Incoming Requests', it shows a request from 'Oyku Yilmaz'.

If you are the creator of the workspace, you cannot quit it. You can only delete the workspace.

If you are one of the collaborators, you can quit a workspace by clicking Quit Workspace in the details page of the workspace.

Files

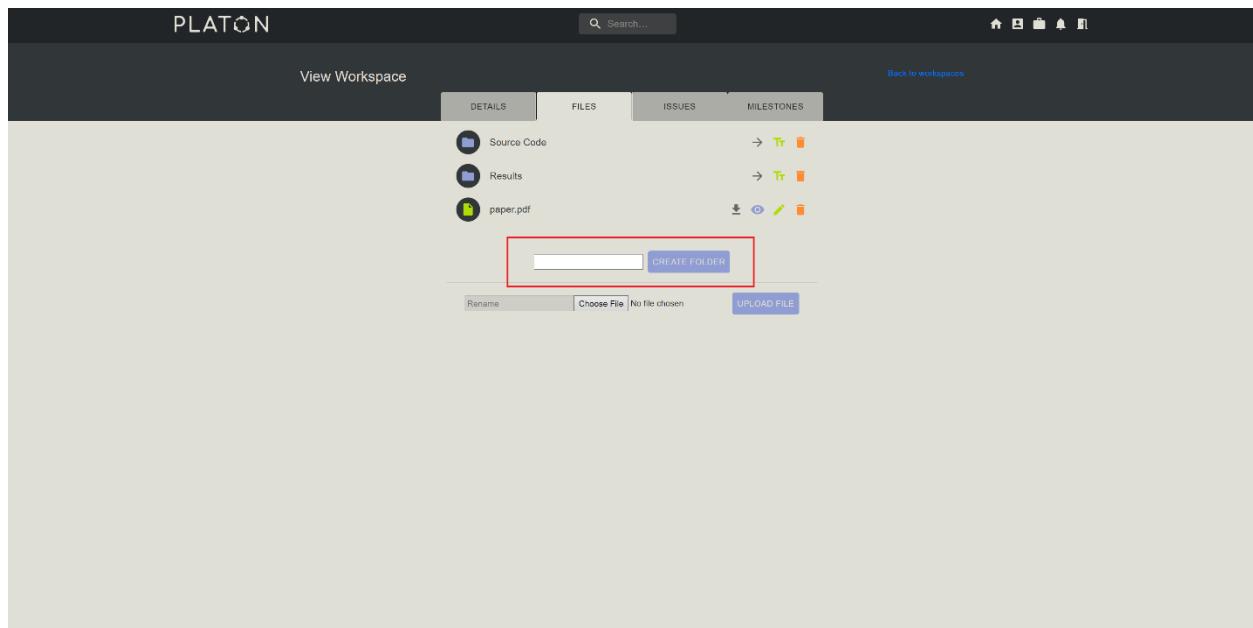
The screenshot shows the 'FILES' section of the workspace. It lists three files:

- Source Code
- Results
- paper.pdf

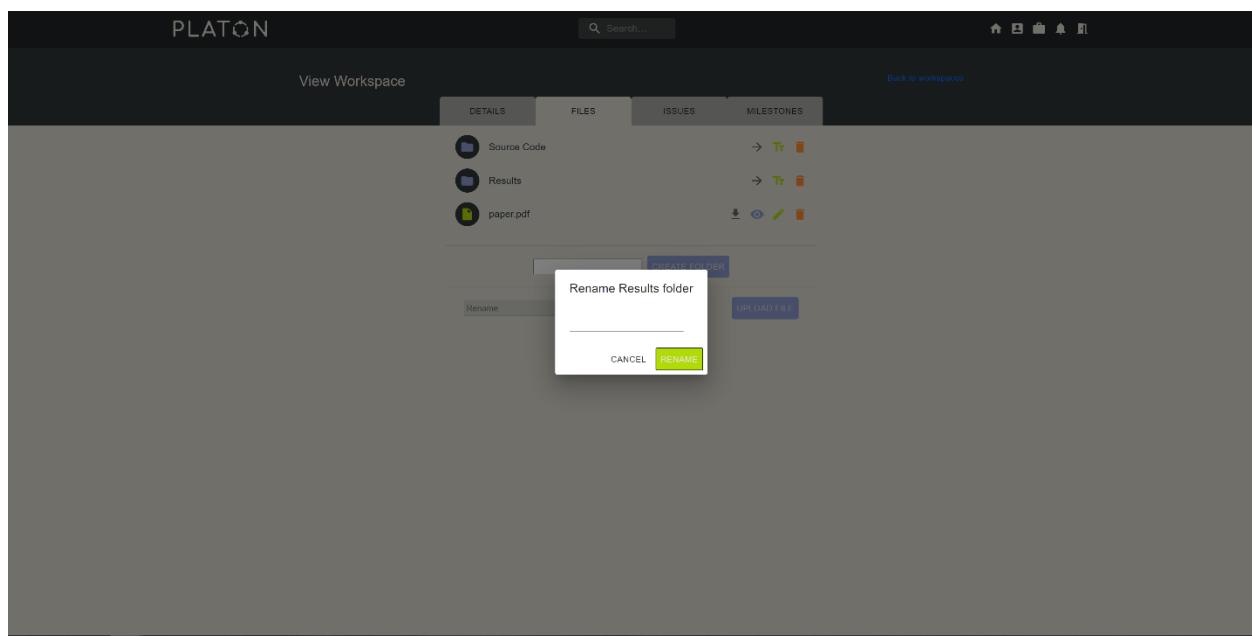
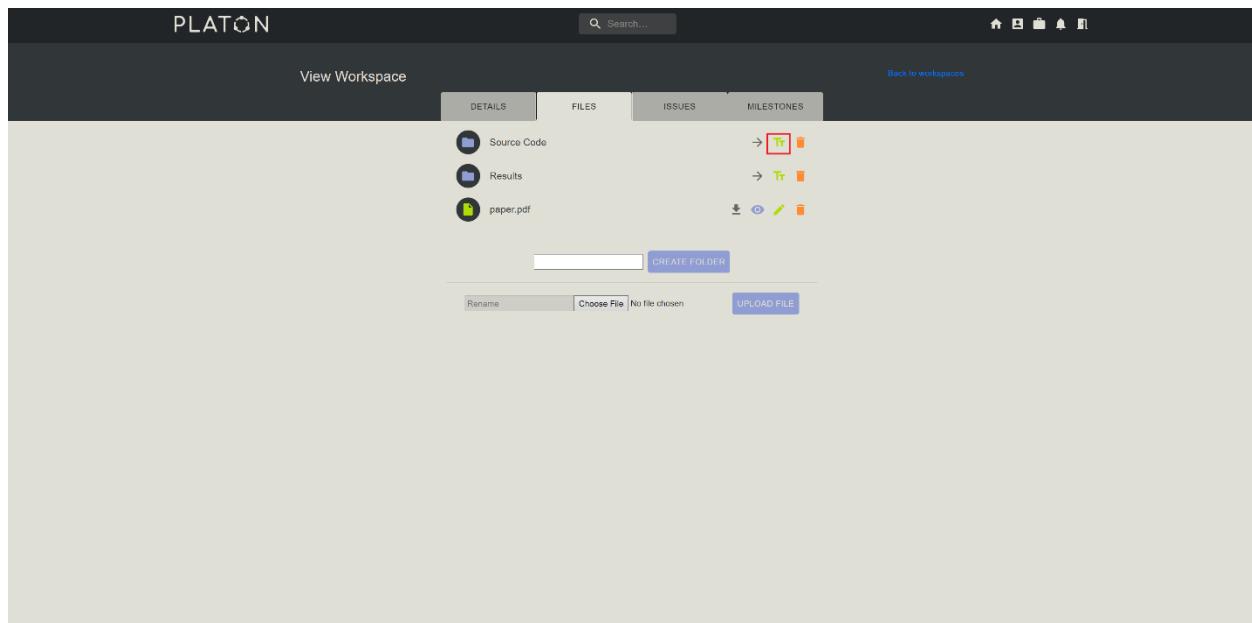
Each file entry includes a preview icon and status indicators (green checkmark, yellow warning, red error). Below the file list, there are buttons for 'CREATE FOLDER', 'Rename', 'Choose File' (with 'No file chosen'), and 'UPLOAD FILE'.

You may add files related to the workspace. You may also create folders for ease of use.

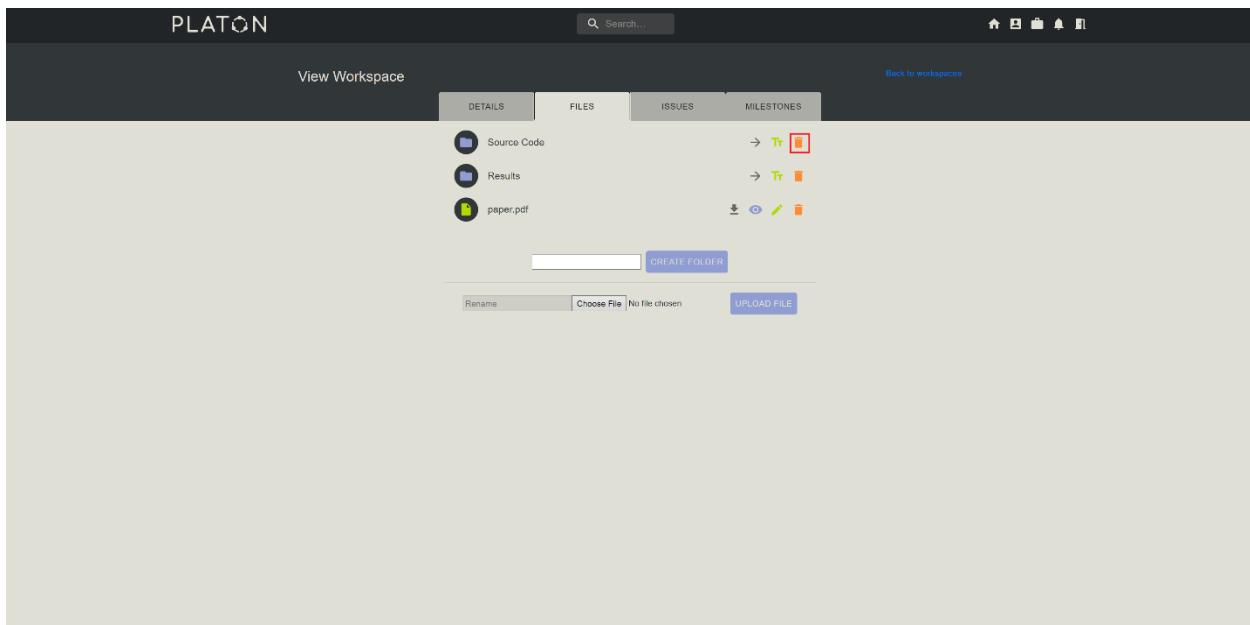
To add or view your files, go to the files section under the workspace page.



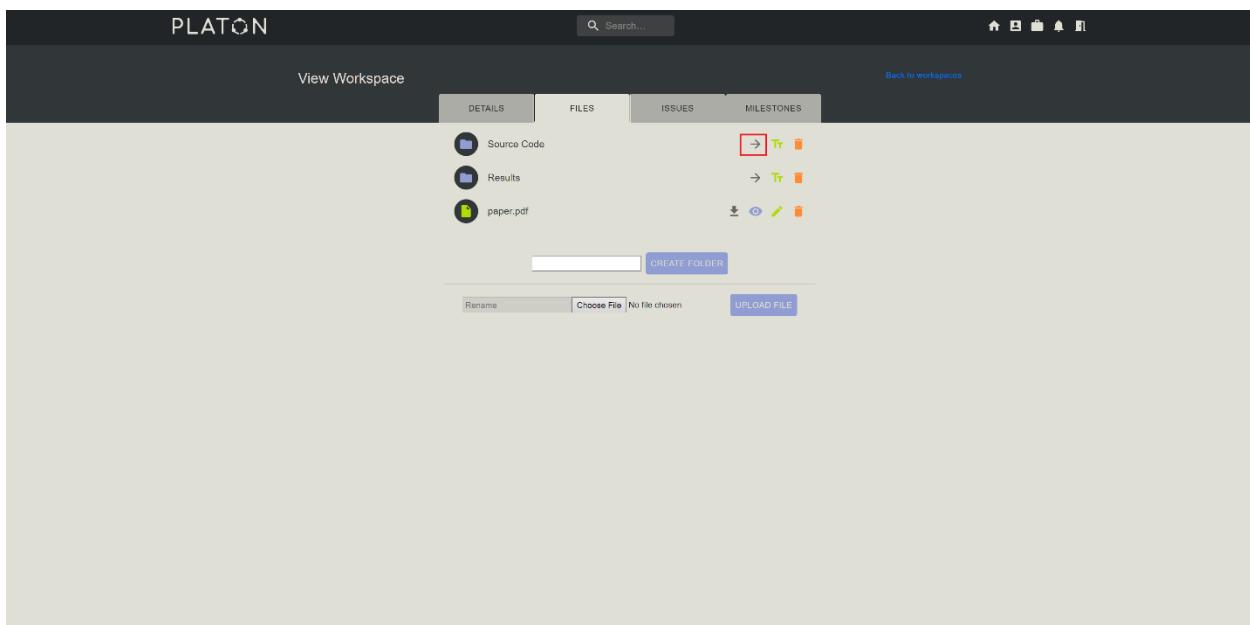
To create a folder, type the name of the folder to the text view on the left of the create folder button. Then click the CREATE FOLDER button.



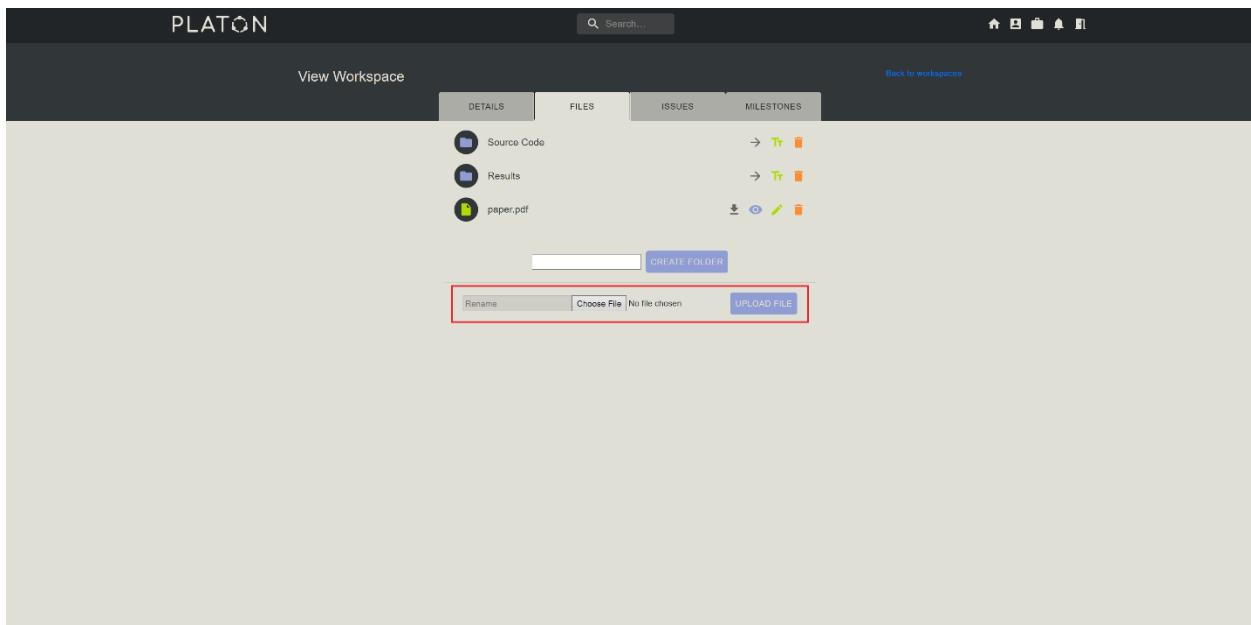
To rename a folder, click the rename button on the right of the desired folder.



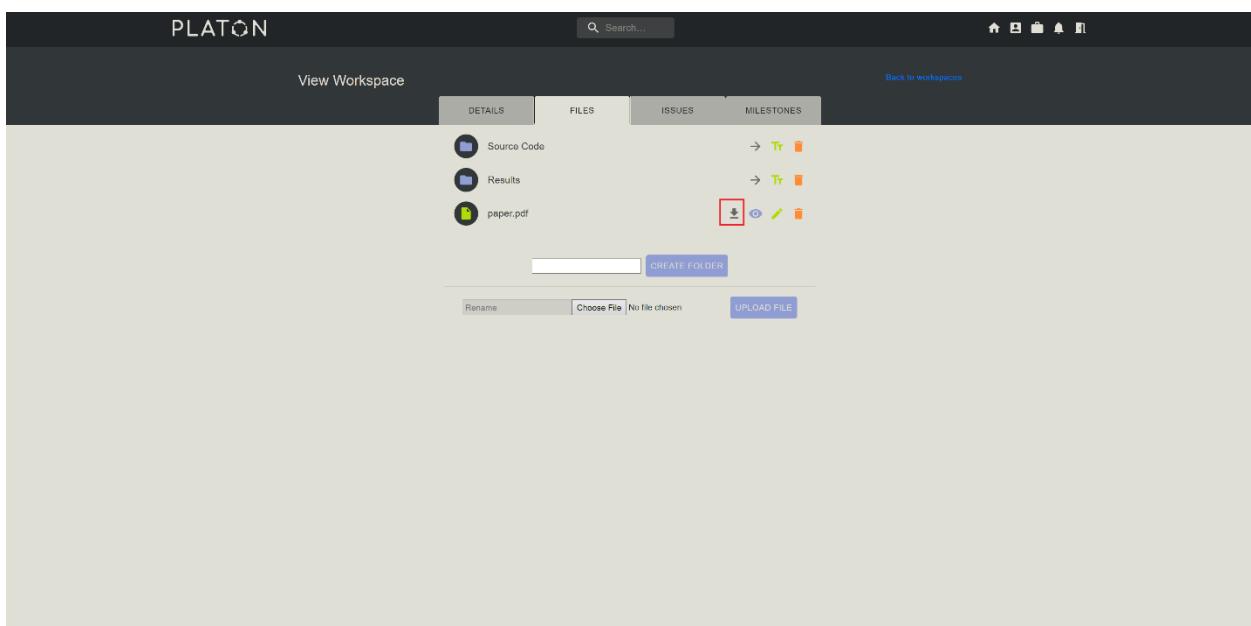
To delete it , click the delete button on the right of the desired folder. This cannot be undone.



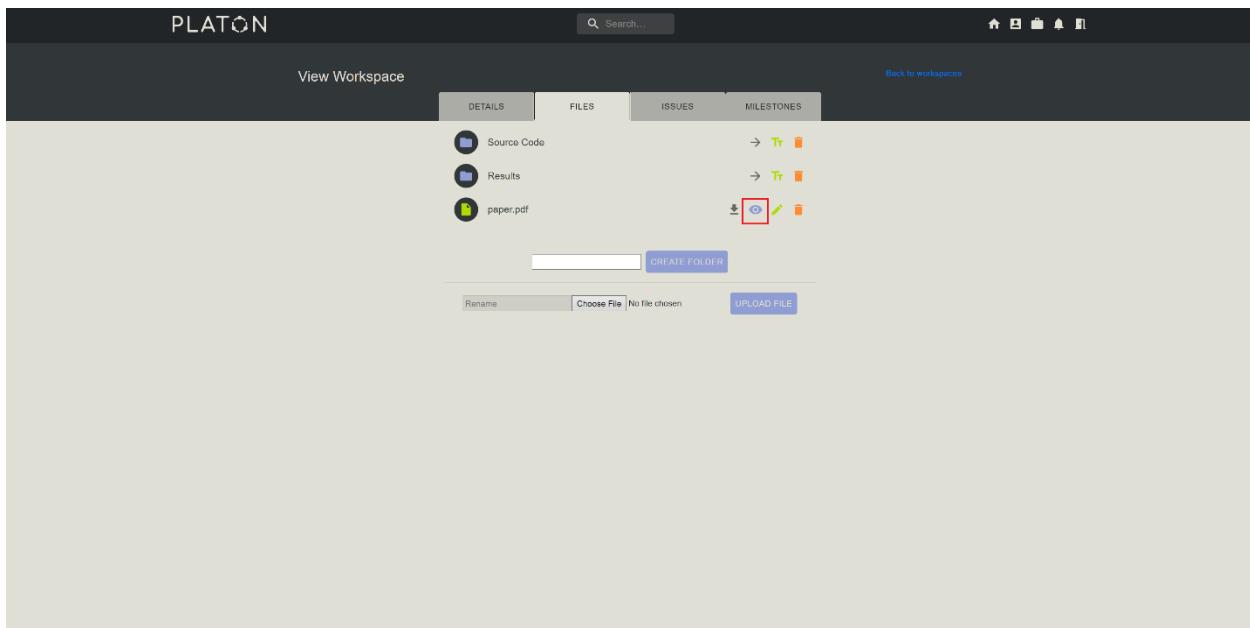
You may check inside a filder by clicking this arrow.



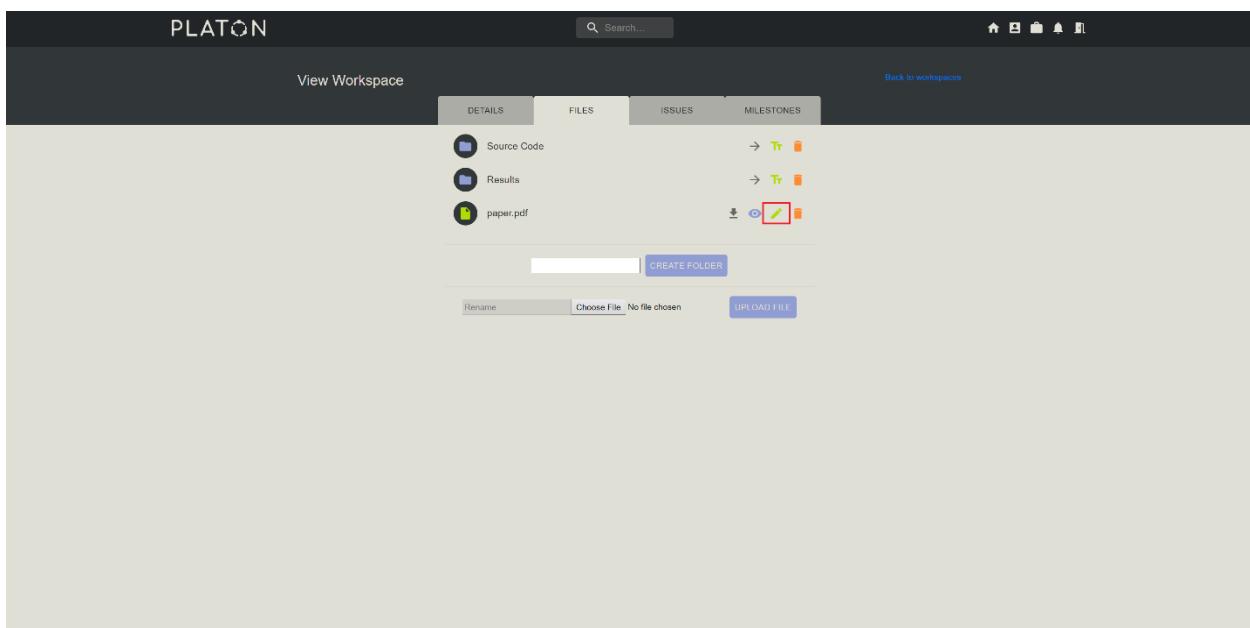
To upload a file, you should click the Choose file button. After you choose your file, you may specify its name. Once you upload it, you will not be able to change its name.



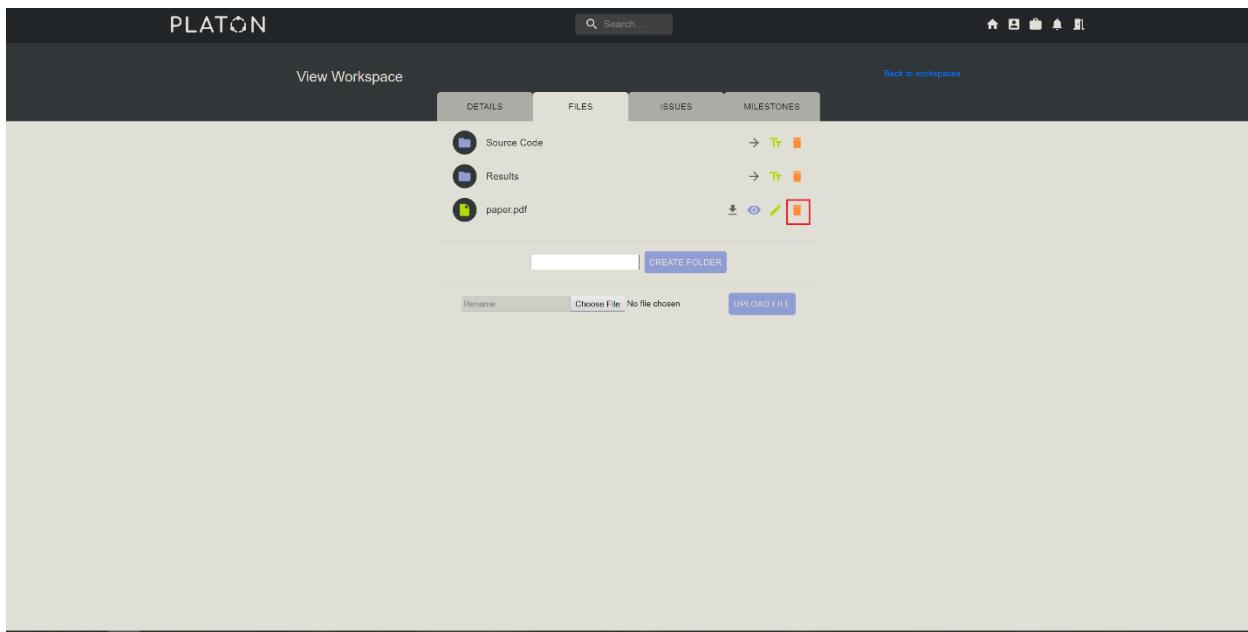
You can download existing files.



You can view the files.

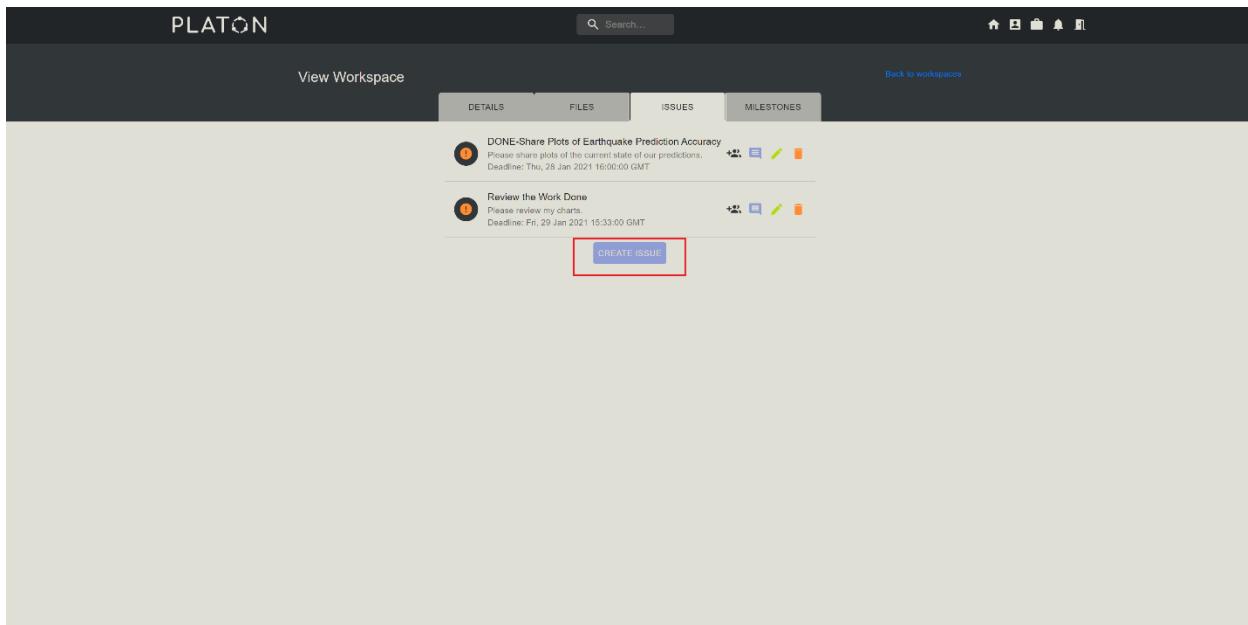


To edit it, click the edit button on the right of the desired file. After editing it, you need to click the save button.

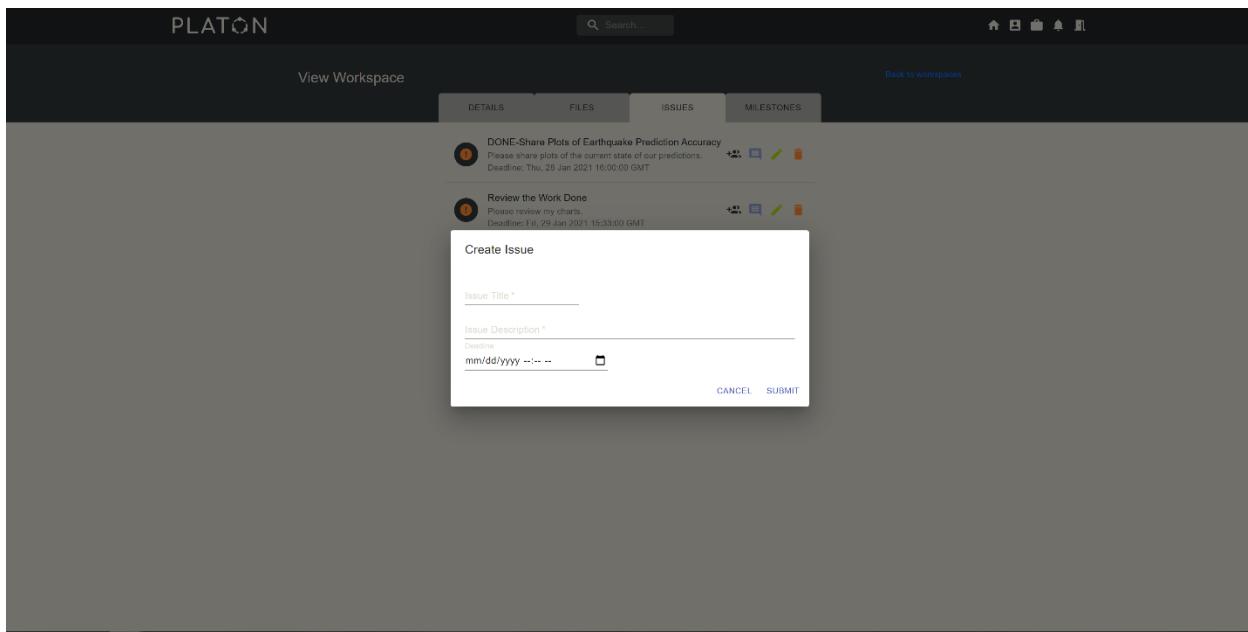


To delete it, click the delete button on the right of the desired file. This cannot be undone.

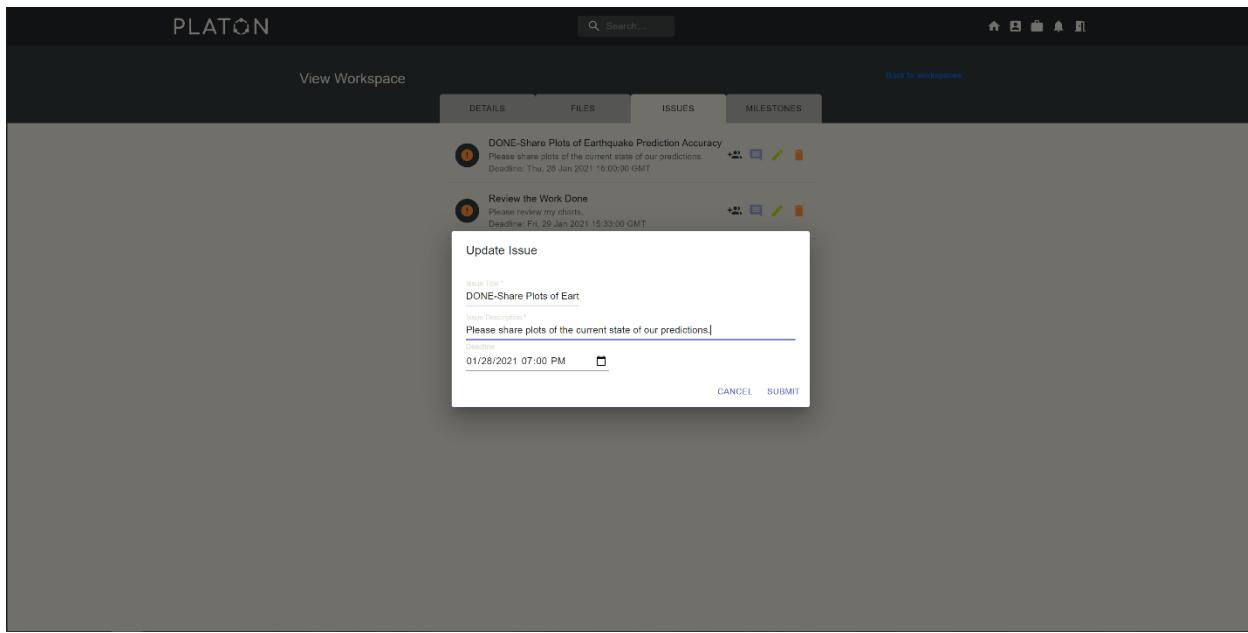
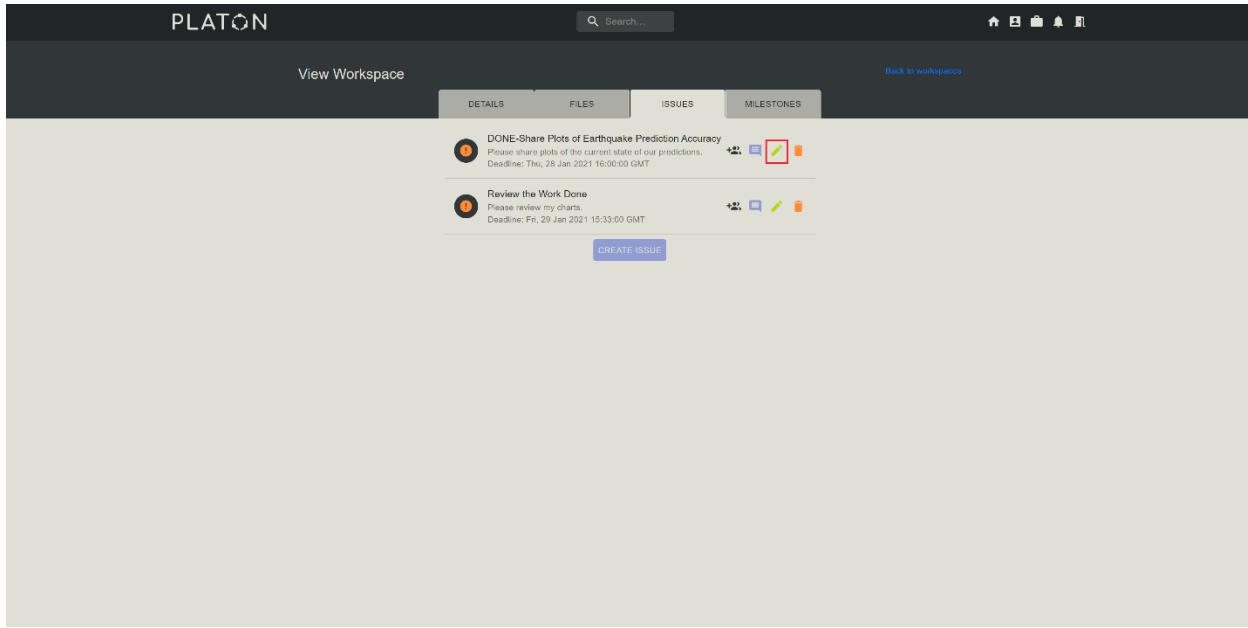
Issues



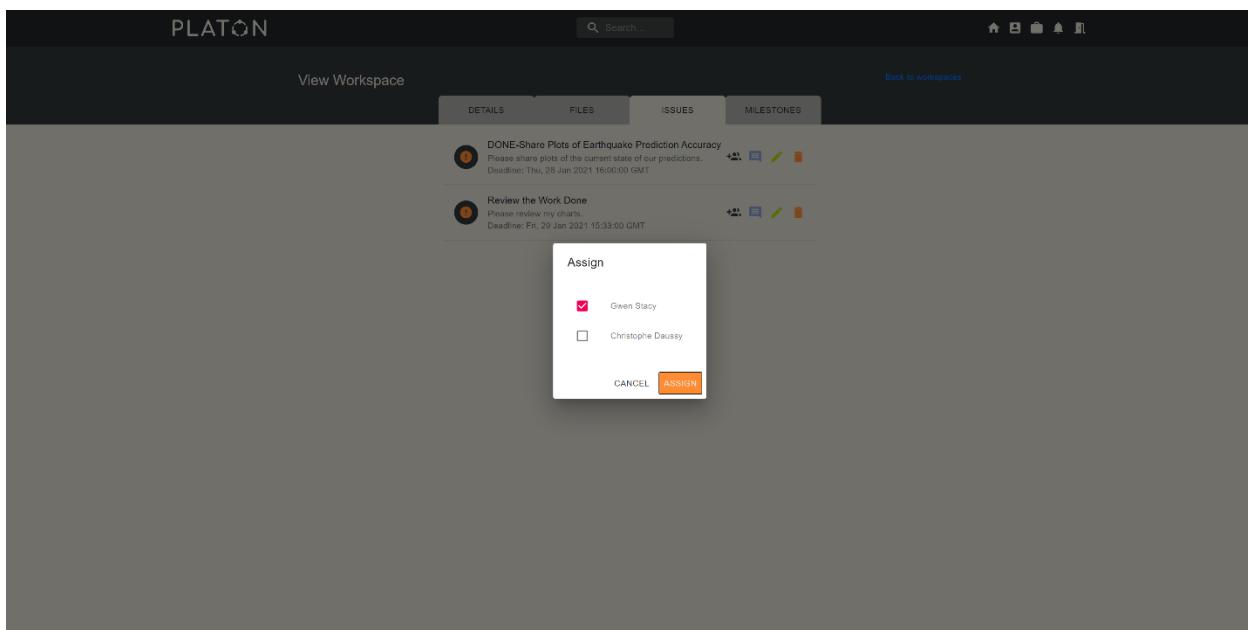
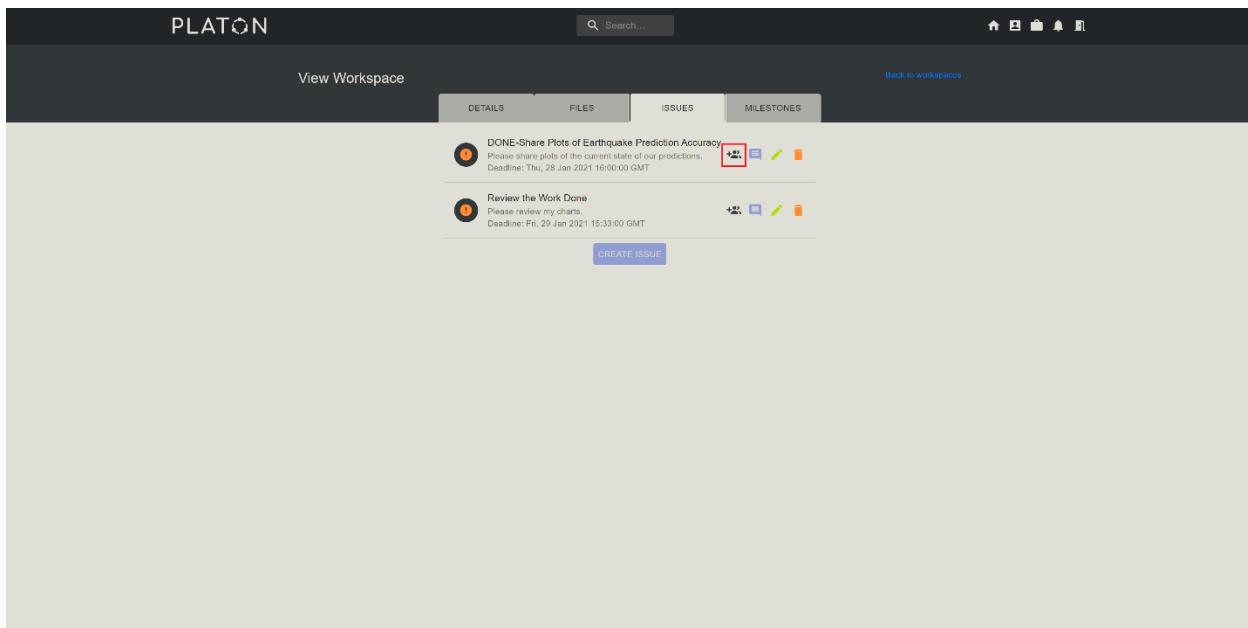
You may create issues to keep track of your future work. To create an issue, go to the issues tab under your workspace's page. To create an issue, click on the button.



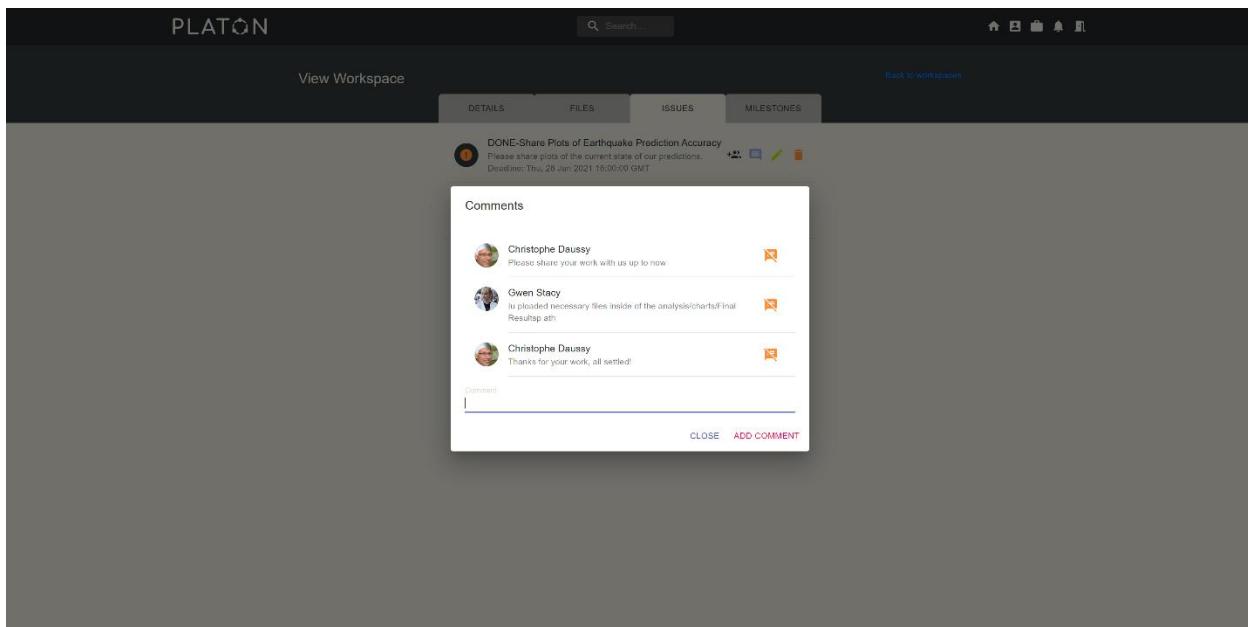
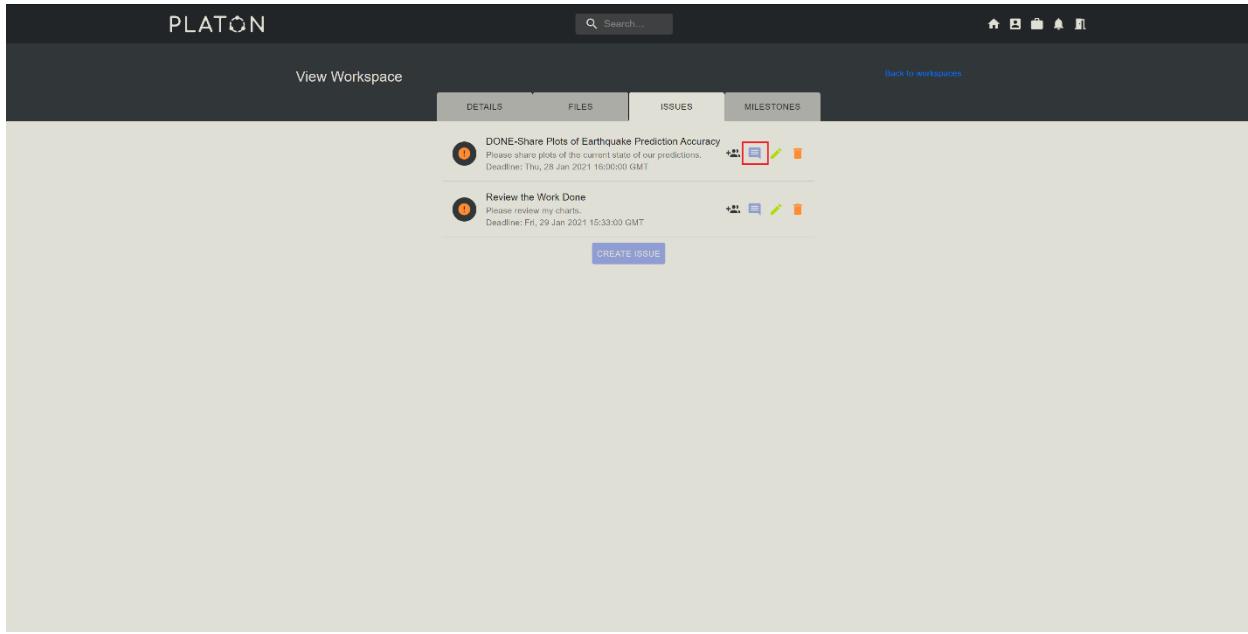
You must provide a title, description and a deadline to the issue to be created.



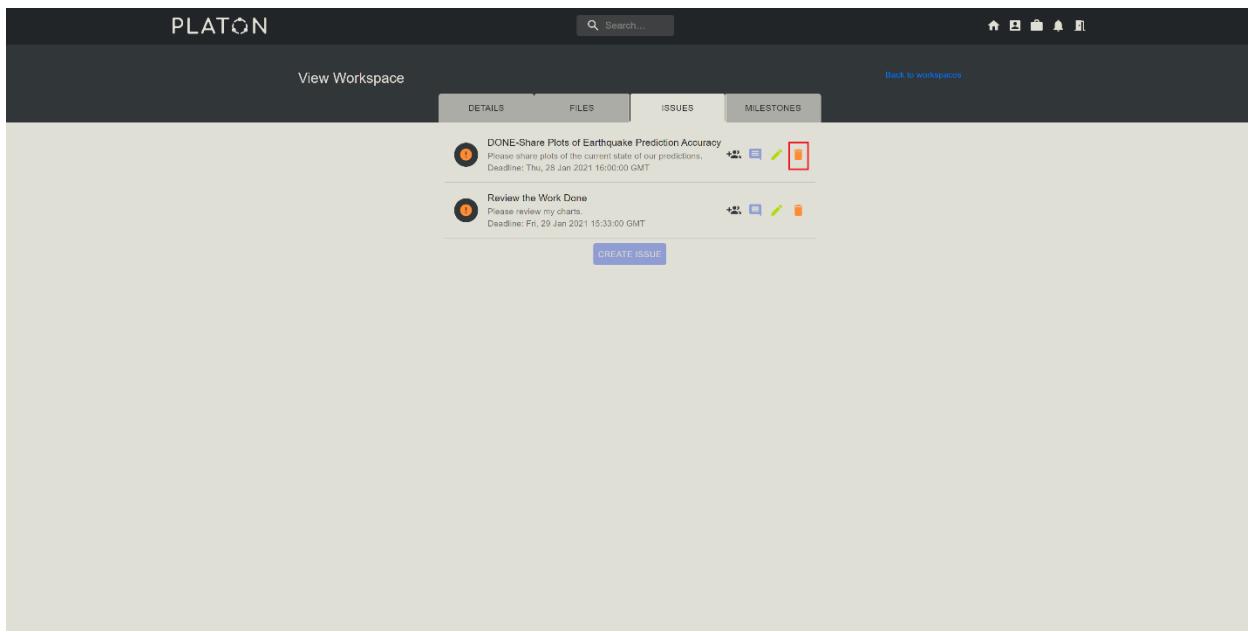
To edit an issue, click on the edit button near that issue.



You can assign collaborators to the specific issue.

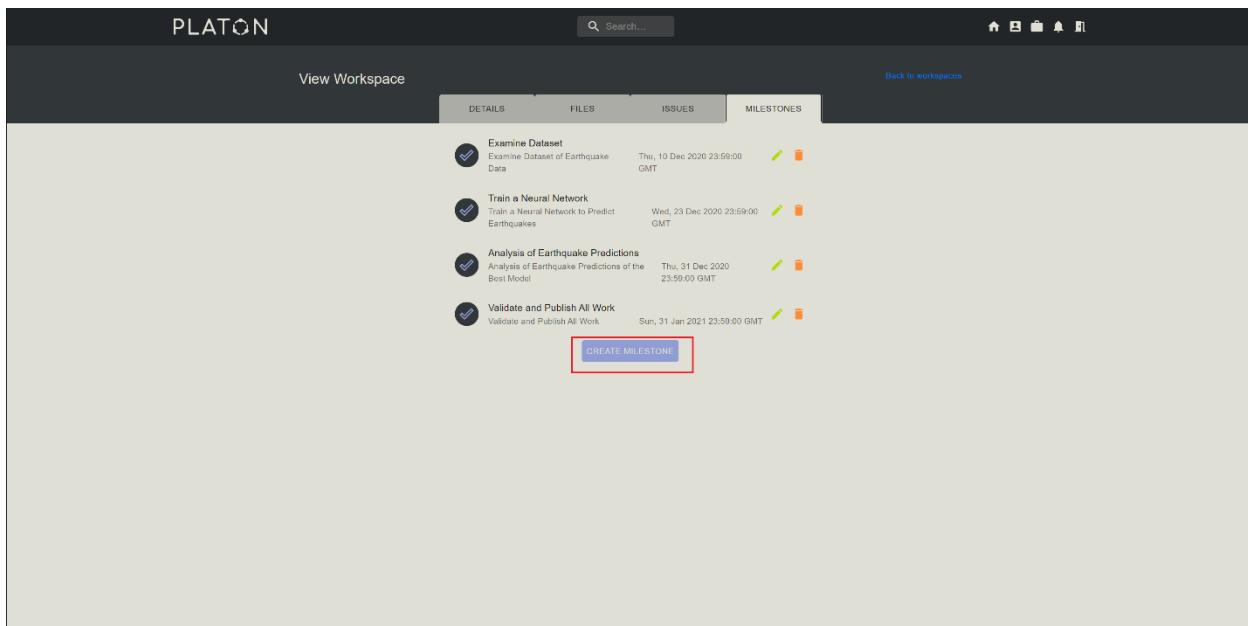


Issues may have comments. You can add comments by clicking this button. You can also delete your previous comments by clicking the orange buttons near that comments.

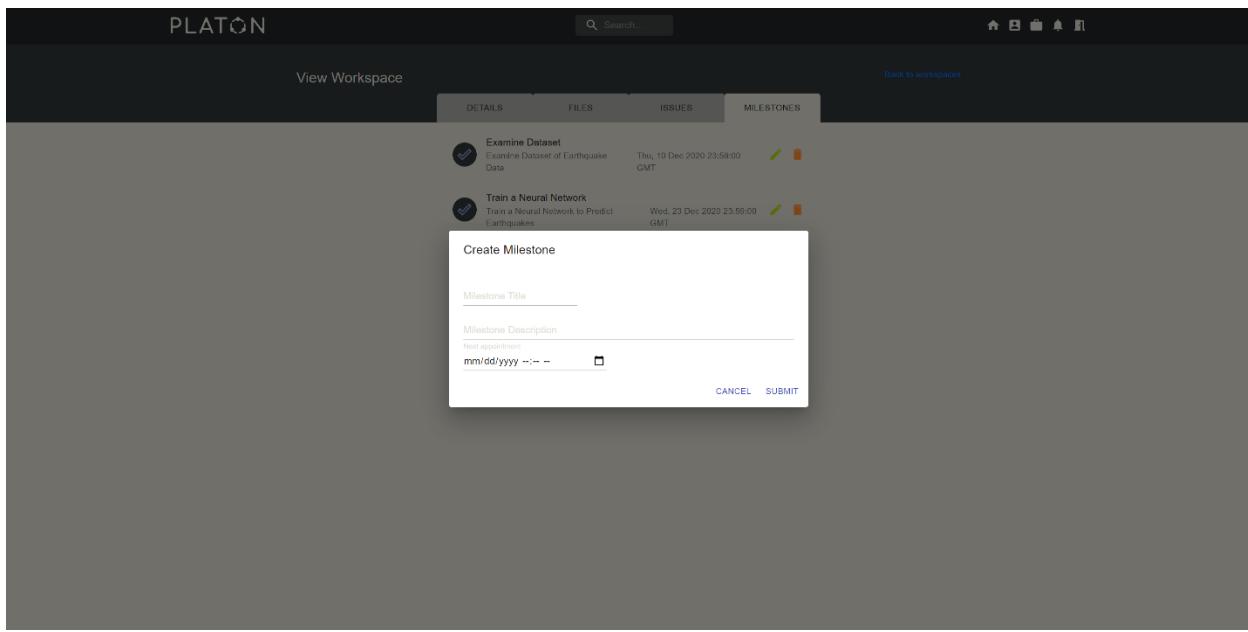


You can also delete an issue. This cannot be undone.

Milestones



You may create milestones to your workspace. To create a milestone, go to the Milestones tab under your workspace's page. To create a milestone, click on the Create Milestone button.



You must provide a title, description and a deadline to the milestone to be created.

PLATON

View Workspace

DETAILS FILES ISSUES MILESTONES Back to workspaces

<input checked="" type="checkbox"/> Examine Dataset Examine Dataset of Earthquake Data	Thu, 10 Dec 2020 23:59:00 GMT	✓ ●
<input checked="" type="checkbox"/> Train a Neural Network Train a Neural Network to Predict Earthquakes	Wed, 23 Dec 2020 23:59:00 GMT	✓ ●
<input checked="" type="checkbox"/> Analysis of Earthquake Predictions Analysis of Earthquake Predictions of the Best Model	Thu, 31 Dec 2020 23:59:00 GMT	✓ ●
<input checked="" type="checkbox"/> Validate and Publish All Work Validate and Publish All Work	Sun, 31 Jan 2021 23:59:00 GMT	✓ ●

[CREATE MILESTONE](#)

PLATON

View Workspace

DETAILS FILES ISSUES MILESTONES Back to workspaces

<input checked="" type="checkbox"/> Examine Dataset Examine Dataset of Earthquake Data	Thu, 10 Dec 2020 23:59:00 GMT	✓ ●
<input checked="" type="checkbox"/> Train a Neural Network Train a Neural Network to Predict Earthquakes	Wed, 23 Dec 2020 23:59:00 GMT	✓ ●
Update Milestone		
Milestone Title <input type="text" value="Examine Dataset"/>		
Milestone Description <input type="text" value="Examine Dataset of Earthquake Data"/>		
Deadline <input type="text" value="12/11/2020 02:59 AM"/>	<input type="button" value="CANCEL"/>	<input type="button" value="SUBMIT"/>

After creating the milestone, you may edit its title, description and the deadline.

The screenshot shows the PLATON workspace interface. At the top, there is a dark header bar with the PLATON logo on the left, a search bar in the center containing the placeholder "Search...", and several icons on the right. Below the header is a navigation bar with tabs: "View Workspace" (selected), "DETAILS", "FILES", "ISSUES", and "MILESTONES". A "Back to workspaces" link is located at the top right of the main content area. The main content area displays a list of milestones:

Milestone	Description	Date	Status
Examine Dataset	Examine Dataset of Earthquake Data	Thu, 10 Dec 2020 23:59:00 GMT	✓ ✖
Train a Neural Network	Train a Neural Network to Predict Earthquakes	Wed, 23 Dec 2020 23:59:00 GMT	✓ ⚠
Analysis of Earthquake Predictions	Analysis of Earthquake Predictions of the Best Model	Thu, 31 Dec 2020 23:59:00 GMT	✓ ⚠
Validate and Publish All Work	Validate and Publish All Work	Sun, 31 Jan 2021 23:59:00 GMT	✓ ⚠

A blue "CREATE MILESTONE" button is located at the bottom right of the milestone list.

To delete a milestone, click on the delete button on the right of the milestone. This cannot be undone.

This is how other workspaces look like. You can only view public workspaces. If a workspace interests you, you may send an application by clicking APPLY WORKSPACE.

If you are the creator of the workspace, you cannot quit it. You have to delete that workspace. If you delete a workspace, you cannot recover it. All the files, milestones, and issues will be deleted. The workspace will automatically disappear in every collaborator's workspace list.

The screenshot shows the PLATON platform interface. At the top, there is a search bar and a navigation menu with icons for home, search, notifications, and account. Below the header, the main content area is divided into sections:

- Workspaces**: A card titled "Predicting Earthquake with DL" with a brief description and two small icons.
- New Workspace**: A button labeled "New Workspace".
- Incoming Workspace Invitations**: A section with a red border containing a message from "Christophe Daussy" inviting the user to join "Discovering Bias in Datasets". It includes a "Decline" (red X) and an "Accept" (green checkmark) button.
- Upcoming Deadlines**: A list of four workspace milestones:
 - Predicting Earthquake with DL, Milestone Deadline: Examine Dataset, 11 Dec 2020 02:59
 - Predicting Earthquake with DL, Milestone Deadline: Train a Neural Network, 24 Dec 2020 02:59
 - Predicting Earthquake with DL, Milestone Deadline: Analysis of Earthquake Predictions, 1 Jan 2021 02:00
 - Predicting Earthquake with DL, Issue Deadline: DONE-Share Plots of Earthquake Prediction Accuracy, 28 Jan 2021 19:00

At the bottom of the "Upcoming Deadlines" section, there is a navigation bar with arrows and the number "1" in a red circle, indicating the current page.

Here you may answer your workspace invitations.

Notifications

PLATON

Trending Workspaces

- Predicting Earthquake with DL
Gwen Stacy - Christophe Daussy
- Medical Information System
Ahmet Dalkı - Bunk Ömür - Umutcan Uut - Oyku Yılmaz - Halil Ümut Özdemir - Burhan Can Akkus - Ertegül Bulut
- Discovering Bias in Datasets
Umutcan Uvt - Oyku Yılmaz - Halli Umut Özdemir - Christophe Daussy

What's Happening?

- Christophe Daussy
Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy created a new folder named 'Data Analysis' in the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy created a new folder named 'Initial Model Analysis' in the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy deleted the file named 'IMG-20190609-WA0001.jpg' in the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy deleted the file named 'Black Heart Emoji On Apple Ios 11 3.jpg' in the workspace 'Predicting Earthquake with DL'.
- can bolukbas
can bolukbas commented and rated Halil Umut Özdemir

Upcoming Events

- ACM Conference on Security and Privacy in Wireless and Mobile Networks
Start Date: Jun 28, 2021 - Jul 1, 2021

Upcoming Deadlines

- Predicting Earthquake with DL
Milestone Deadline: Examine Dataset
11 Dec 2020 02:59
- Predicting Earthquake with DL
Milestone Deadline: Train a Neural Network
24 Dec 2020 02:59
- Predicting Earthquake with DL
Milestone Deadline: Analysis of Earthquake Predictions
1 Jan 2021 02:59

PLATON

Trending Workspaces

- Predicting Earthquake with DL
Gwen Stacy - Christophe Daussy
- Medical Information System
Ahmet Dalkı - Bunk Ömür - Umutcan Uut - Oyku Yılmaz - Halil Ümut Özdemir - Burhan Can Akkus - Ertegül Bulut
- Discovering Bias in Datasets
Umutcan Uvt - Oyku Yılmaz - Halli Umut Özdemir - Christophe Daussy

What's Happening?

- Christophe Daussy
Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy created a new folder named 'Data Analysis' in the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy created a new folder named 'Initial Model Analysis' in the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy deleted the file named 'IMG-20190609-WA0001.jpg' in the workspace 'Predicting Earthquake with DL'.
- Christophe Daussy
Christophe Daussy deleted the file named 'Black Heart Emoji On Apple Ios 11 3.jpg' in the workspace 'Predicting Earthquake with DL'.
- can bolukbas
can bolukbas commented and rated Halil Umut Özdemir
- Oyku Yılmaz
Oyku Yılmaz commented and rated Halil Umut Özdemir
- Christophe Daussy
Christophe Daussy deleted the folder named 'Charts' in the workspace 'Predicting Earthquake with DL'.

Upcoming Deadlines

- Predicting Earthquake with DL
Milestone Deadline: Examine Dataset
11 Dec 2020 02:59
- Predicting Earthquake with DL
Milestone Deadline: Train a Neural Network
24 Dec 2020 02:59
- Predicting Earthquake with DL
Milestone Deadline: Analysis of Earthquake Predictions
1 Jan 2021 02:59

Bunk Ömür sent you a follow request
1/27/2021

Oyku Yılmaz sent you a follow request
1/27/2021

31 new upcoming events have been added.
1/27/2021

To see your notifications, click the button. You may click on the pages to load more notifications.

Home Page

Home page consists of mainly four parts: Upcoming events, Trending Projects, Personal Reminder, and the Activity Stream.

The screenshot shows the PLATON platform's home page with the following sections:

- Trending Workspaces**: A section highlighted with a red border, containing three workspace cards:
 - Predicting Earthquake with DL by Gwen Stacy - Christophe Daussy
 - Medical Information System by Ahmet Dadak - Burak Ömür - Umutcan Ünüt - Oykù Yılmaz - Halil Umut Özdemir - Burhan Can Akkus - Ertrugrul Bulbul
 - Discovering Bias in Datasets by Umutcan Ünüt - Oykù Yılmaz - Halil Umut Özdemir - Christophe DaussyA navigation bar below the cards shows pages 1, 2, 3, 4, and >.
- What's Happening?**: A list of recent activity items:
 - Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'
 - Christophe Daussy created a new folder named 'Data Analysis' in the workspace 'Predicting Earthquake with DL'
 - Christophe Daussy created a new folder named 'Initial Model Analysis' in the workspace 'Predicting Earthquake with DL'
 - Christophe Daussy deleted the file named 'IMG-20190609-WA0001.jpg' in the workspace 'Predicting Earthquake with DL'
 - Christophe Daussy deleted the file named 'Black Heart Emoji On Apple Ios 11 3.jpg' in the workspace 'Predicting Earthquake with DL'
 - can bolukbas commented and rated Halil Umut Özdemir
- Upcoming Events**: A list of upcoming events:
 - ACM Conference on Security and Privacy in Wireless and Mobile Networks (Start Date: Jun 28, 2021 - Jul 1, 2021)A disclaimer at the bottom states: "Disclaimer: Data is provided by CFP API."
- Upcoming Deadlines**: A list of upcoming deadlines:
 - Predicting Earthquake with DL: Milestone Deadline: Examine Dataset (11 Dec 2020 02:59)
 - Predicting Earthquake with DL: Milestone Deadline: Train a Neural Network (24 Dec 2020 02:59)
 - Predicting Earthquake with DL: Milestone Deadline: Analysis of Earthquake Predictions (1 Jan 2021 02:59)A navigation bar below the deadlines shows pages <, 1, 2, >.

Under this section, trending public workspaces are listed. You may see its details by clicking on the name of the project.

The screenshot shows the PLATON interface with a dark theme. On the left, there's a sidebar titled "Trending Workspaces" listing three workspaces: "Predicting Earthquake with DL", "Medical Information System", and "Discovering Bias in Datasets". The main area features a "What's Happening?" feed with a red border around it. This feed displays activity items from user Christophe Daussy, such as uploading files and creating folders, and a comment from user can bolukbas. To the right, there are sections for "Upcoming Events" and "Upcoming Deadlines", each with a green header and a list of items. A navigation bar at the top includes icons for home, search, and notifications.

Here is the activity stream. You can see general activities of the users you follow.

This screenshot is identical to the one above, displaying the same activity stream, workspace trends, and upcoming events. The layout and data points (Christophe Daussy's actions and the green event box) are identical to the first screenshot.

Upcoming Conferences, Journal Special Issues, Submission Deadlines and CFP (call for papers) documents. These are provided by CFP API. If you search for an upcoming event, you may visit its website by clicking on the search result.

The screenshot shows the PLATON application interface. At the top, there is a search bar and a navigation menu with icons for home, file, settings, notifications, and help. Below the header, there are three main sections: "Trending Workspaces", "What's Happening?", and "Upcoming Events".

- Trending Workspaces:** Displays three workspace cards:
 - Predicting Earthquake with DL (Gwen Stacy - Christophe Daussy)
 - Medical Information System (Ahmet Dadak - Burak Ömür - Umutcan Üvut - Oykù Yılmaz - Halil Ümit Özdemir - Burhan Can Akkus - Ertegun Bülbül)
 - Discovering Bias in Datasets (Umutcan Üvut - Oykù Yılmaz - Halli Umut Özdemir - Christophe Daussy)
- What's Happening?**: A timeline of recent activity:
 - Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy created a new folder named 'Data Analysis' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy deleted the file named 'IMG-20190609-WA0001.jpg' in the workspace 'Predicting Earthquake with DL'.
 - Christophe Daussy deleted the file named 'Black Heart Emoji On Apple Ios 11 3.jpg' in the workspace 'Predicting Earthquake with DL'.
 - can bolukbas commented and rated Halil Umut Özdemir
- Upcoming Events:** A list of events with a red border around the "Upcoming Deadlines" section:
 - ACM Conference on Security and Privacy in Wireless and Mobile Networks (Start Date: Jun 28, 2021 - Jul 1, 2021)
 - Upcoming Deadlines** (highlighted):
 - Predicting Earthquake with DL (Milestone Deadline: Examine Dataset 11 Dec 2020 02:59)
 - Predicting Earthquake with DL (Milestone Deadline: Train a Neural Network 24 Dec 2020 02:59)
 - Predicting Earthquake with DL (Milestone Deadline: Analysis of Earthquake Predictions 1 Jan 2021 02:59)

Personal reminder consists of the deadlines of issues, milestones, and workspaces which you are a collaborator of. They are listed in ascending order.

Search

The screenshot shows the PLATON application interface, similar to the previous one but with the search bar highlighted by a red border. The layout includes a header with a search bar and navigation icons, followed by the same three sections: "Trending Workspaces", "What's Happening?", and "Upcoming Events".

- Trending Workspaces:** Same as the first screenshot.
- What's Happening?**: Same as the first screenshot.
- Upcoming Events:** Same as the first screenshot.

Platon supports semantic search. Semantic search is a data searching technique in which a search query aims to not only find keywords but to determine the intent and contextual meaning of the words a person is using for search.

The screenshot shows the PLATON interface with a search bar containing 'bur'. The search results are displayed in a grid:

- Alperen Divriklioglu**: Profile picture of a man in a blue shirt, job title 'Job'.
- Burak Ömür**: Profile picture of two people, job title 'Job'.
- Burhan Can Akkus**: Profile picture with the name 'NACHTLEBEN' above it.
- Halil Umut Özdemir**: Profile picture with the acronym 'ATO'.

User Search

The screenshot shows the PLATON interface with a search bar containing 'estimating'. The search results are displayed in a grid:

- Automated Conduction for Symphony Orchestras**: Profile picture with the acronym 'ATO', description about conductor management, and author 'by Burhan Can Akkus'.
- Trajectory Prediction for Autonomous Cars**: Profile picture with the acronym 'ATO', description about predicting car trajectories, and author 'by Halil Umut Özdemir'.
- Discovering Bias in Datasets**: Profile picture with the acronym 'ATO', description about uncovering biases in datasets, and author 'by Christophe Daussy'.
- TREC COVID**: Profile picture with the acronym 'ATO', description about COVID-19 research, and author 'by Researchers, clinicians, and policy makers involved with the response to COVID-19'.

Workspace Search

The screenshot shows the PLATON search interface with a dark theme. On the left, there is a sidebar with search filters:

- Search bar: "ai"
- User checkbox
- Workspace checkbox
- Upcoming Events
- Tag Search checkbox
- Start Date From: mm/dd/yyyy -> --
- Start Date To: mm/dd/yyyy --> --
- End Date From: mm/dd/yyyy --> --
- End Date To: mm/dd/yyyy --> --
- Sorting Criteria dropdown

On the right, there is a list of five upcoming events, each represented by a blue card with a person icon:

- International Conference on Computer, Information and Telecommunication Systems
Jul 29, 2021 - Jul 31, 2021
- International Conference on Computer, Information and Telecommunication Systems
Jul 29, 2021 - Jul 31, 2021
- International Conference on Recent Advances in Railway Engineering
May 17, 2021 - May 18, 2021
- (ONLINE) POSTMEMORY AND THE CONTEMPORARY WORLD 2nd International Interdisciplinary Conference
Feb 25, 2021 - Feb 26, 2021
- 10th EAI International Conference on Context-Aware Systems and Applications (Springer, Scopus, ISI, Ei, more)
Oct 28, 2021 - Oct 29, 2021

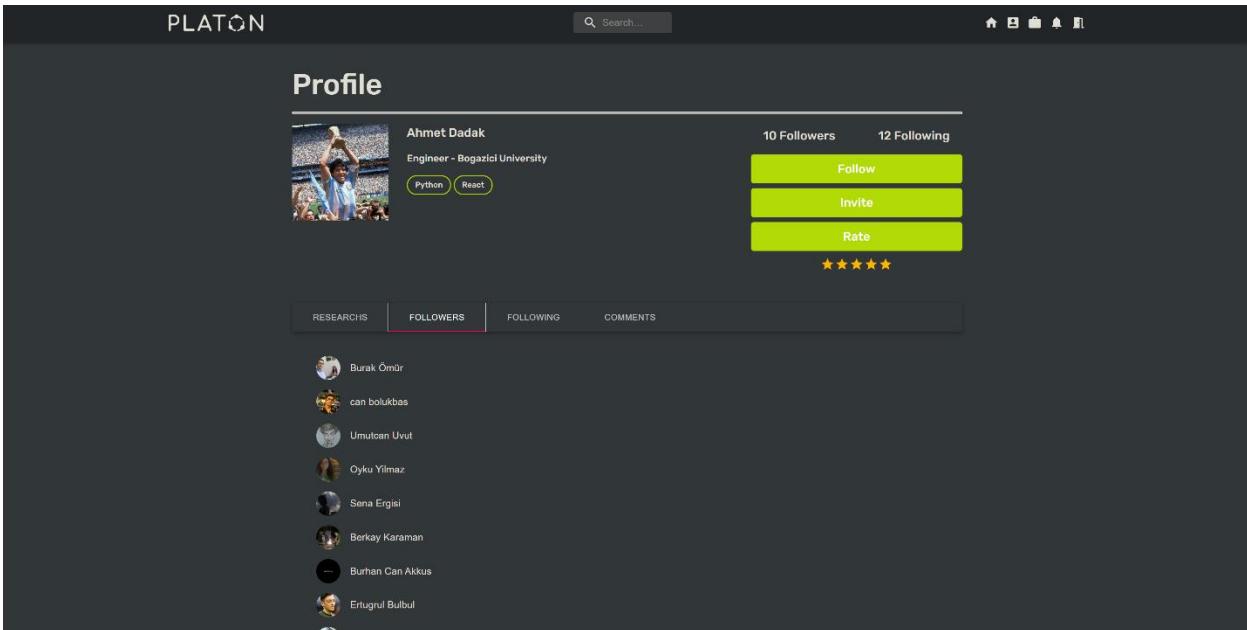
Upcoming Event Search

You may search for users, workspaces, and upcoming events.

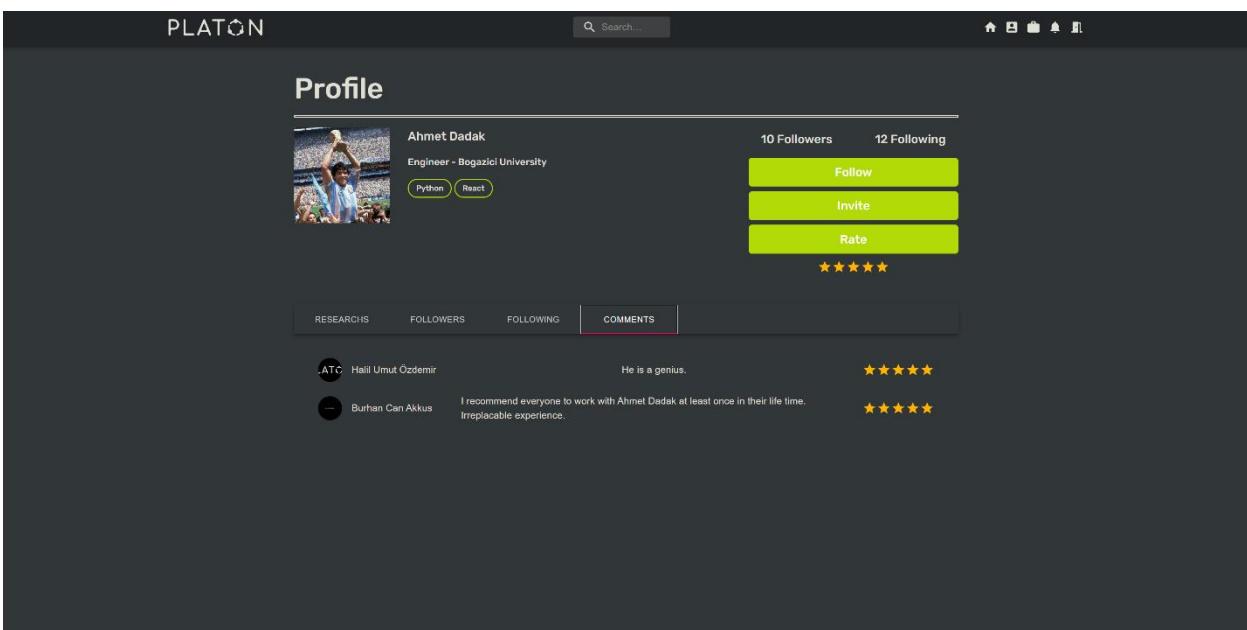
You may use the filters related to every search type to narrow down your search.

Only public workspaces will be listed in the workspace search.

Other Users' Profiles

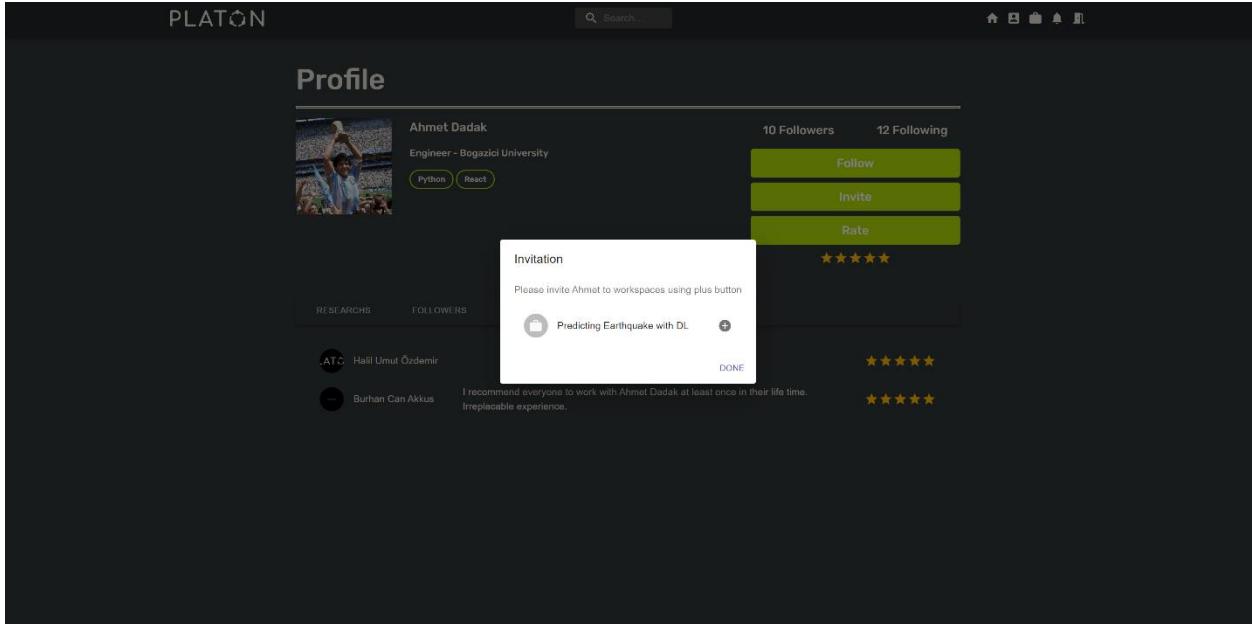


This screenshot shows a user profile page for 'Ahmet Dadak' on the PLATON platform. The profile header features a photo of a person in a blue shirt, the name 'Ahmet Dadak', and the title 'Engineer - Bogazici University'. Below the name are two skill badges: 'Python' and 'React'. To the right, it shows '10 Followers' and '12 Following'. A row of three green buttons labeled 'Follow', 'Invite', and 'Rate' is present, along with a five-star rating icon. Below these are tabs for 'RESEARCHES', 'FOLLOWERS', 'FOLLOWING', and 'COMMENTS'. Under the 'FOLLOWERS' tab, a list of 10 users is shown, each with a small profile picture and their name: Burak Ömür, can bolukbas, Umutcan Uvut, Oyku Yılmaz, Sena Ergisi, Berkay Karaman, Burhan Can Akkus, and Ertegun Bulbul.



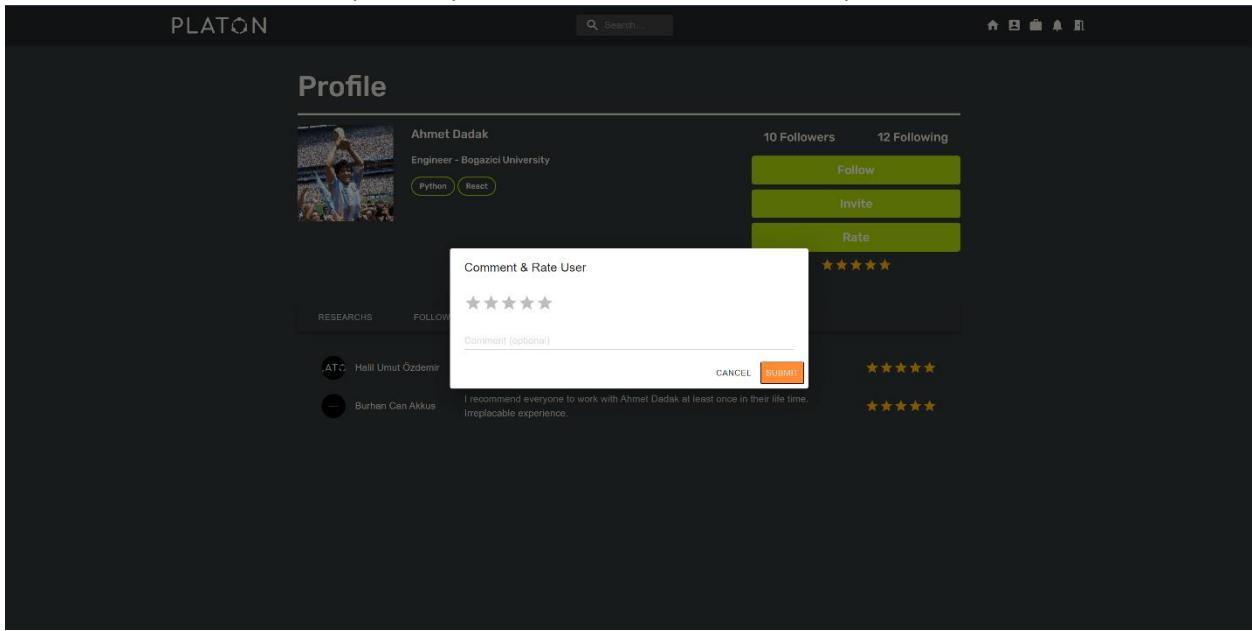
This screenshot shows the same user profile for 'Ahmet Dadak' on PLATON, but with additional comments displayed under the 'COMMENTS' tab. The first comment is from a user with the handle 'LATC' (represented by a black circle icon) and the name 'Halil Umut Özdemir', which reads: 'He is a genius.' The second comment is from 'Burhan Can Akkus' (black circle icon), which reads: 'I recommend everyone to work with Ahmet Dadak at least once in their life time. Irreplaceable experience.' Both comments have a five-star rating icon below them.

This is how a public profile seems. You can see the research, followers, following and comments of the user.



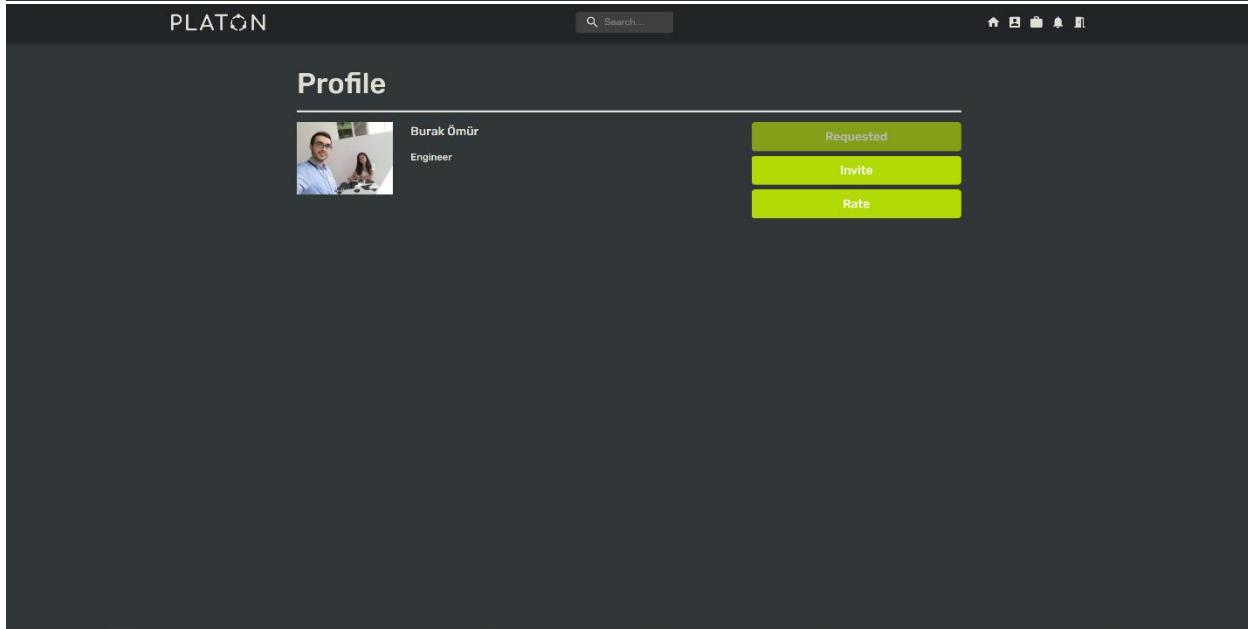
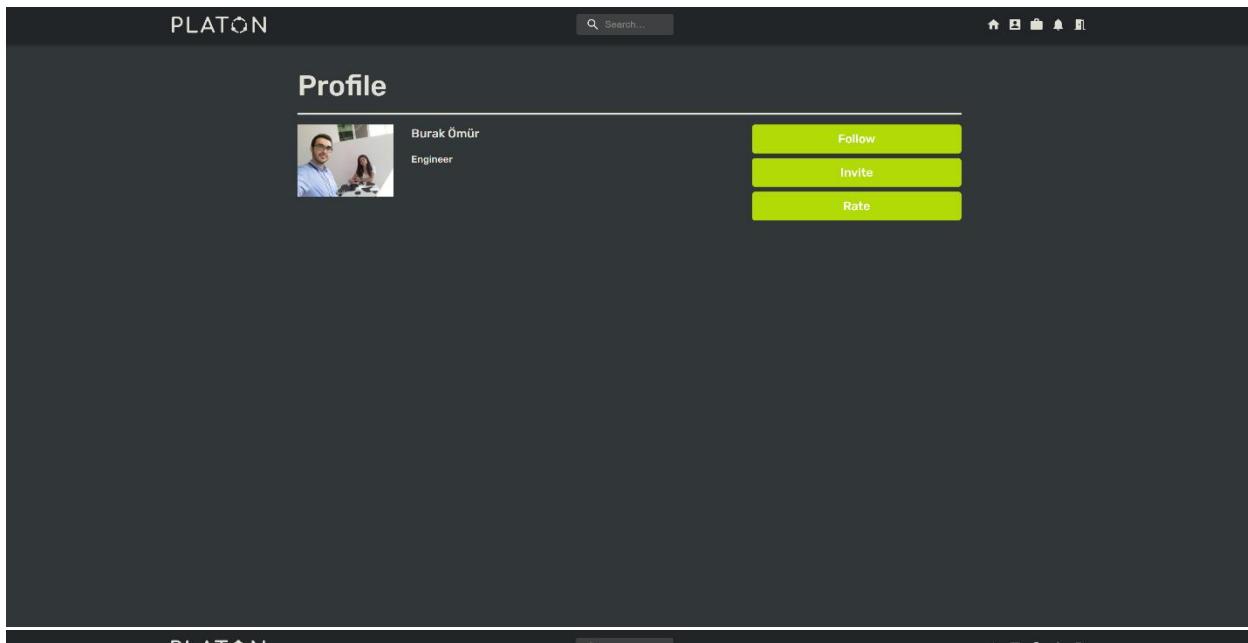
The screenshot shows a dark-themed PLATON profile page for a user named Ahmet Dadak. At the top, there's a search bar and a navigation menu with icons for home, search, notifications, and more. Below the header, the profile section includes a thumbnail image of Ahmet Dadak, his name, title ('Engineer - Bogazici University'), and skills ('Python', 'React'). It also shows his follower count (10 Followers) and the number of users he follows (12 Following). To the right of these stats are three green buttons: 'Follow', 'Invite', and 'Rate'. Below these buttons are five yellow star ratings. A central modal window titled 'Invitation' contains the text 'Please invite Ahmet to workspaces using plus button' and a 'DONE' button. In the background, there are sections for 'RESEARCHES' and 'FOLLOWERS', each showing a list of users with their profile pictures and names.

You can invite users to workspaces if you are the creator of that workspace.



This screenshot shows the same PLATON profile page for Ahmet Dadak. The interface is identical to the previous one, with the addition of a central modal window titled 'Comment & Rate User'. This modal features a five-star rating system, a text input field labeled 'Comment (optional)', and two buttons at the bottom: 'CANCEL' and 'SUBMIT'. The rest of the page, including the user's profile picture, name, title, skills, follower counts, and the 'Follow', 'Invite', and 'Rate' buttons, remains visible in the background.

You can comment and rate profiles if you have collaborated with them before.

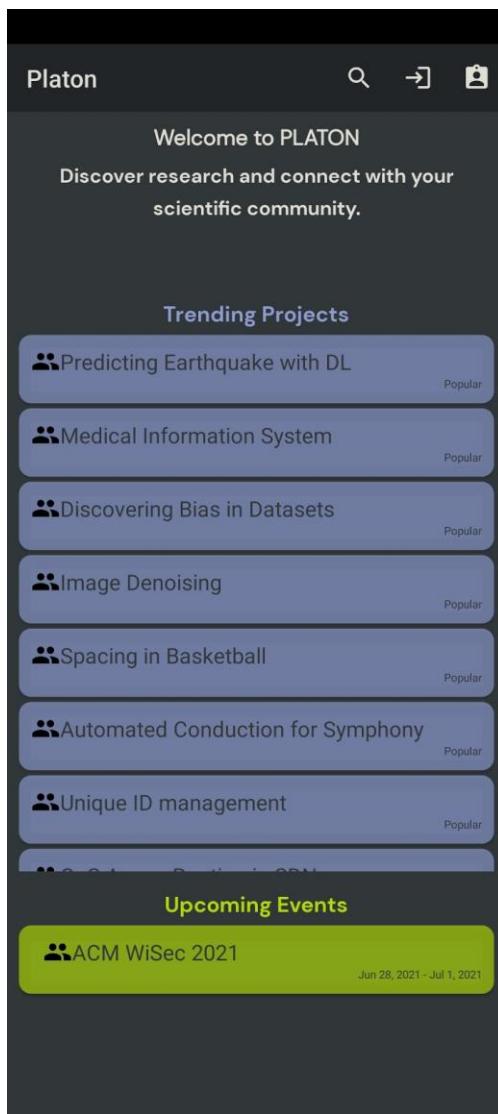


This is how a private profile looks like. You need to send them a follow request before you can see their profile. You can still rate them, or invite them to workspace.

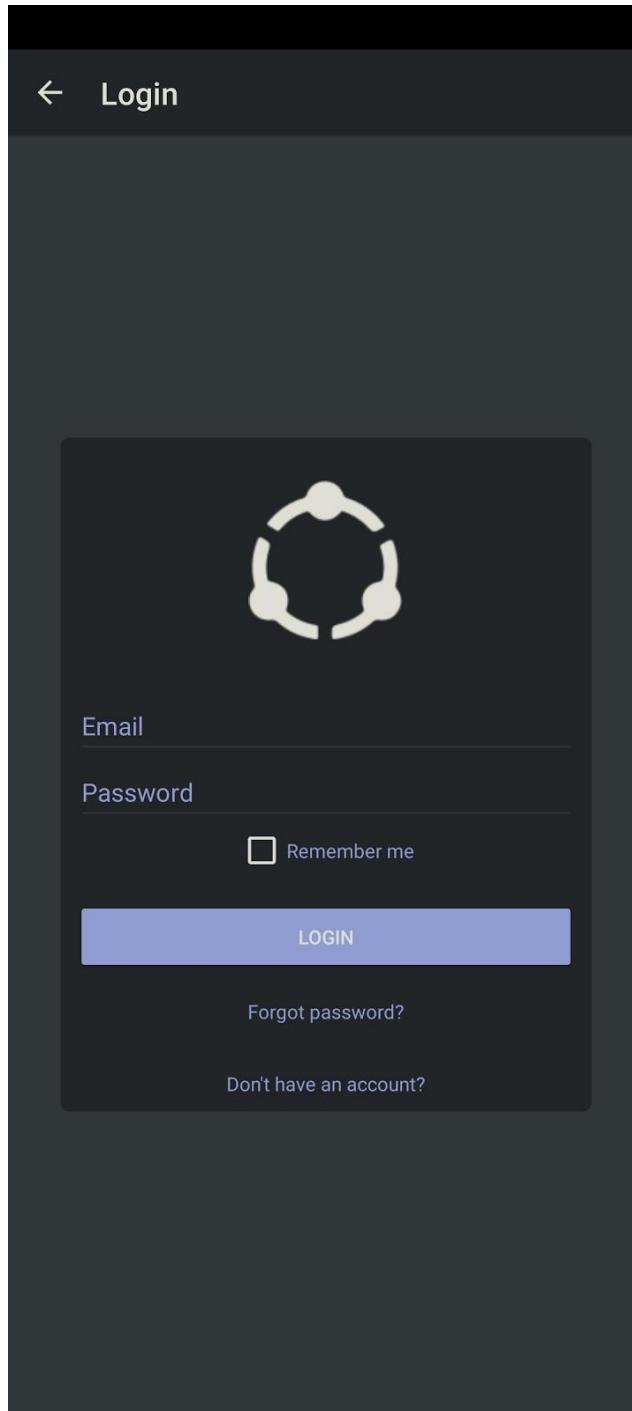
ANDROID

Platon is an academic collaboration platform that unites everyone that has interest in being part of a research project. It is prepared by 7th group for CMPE 352/451 course, Bogazici University students.

For Guests:

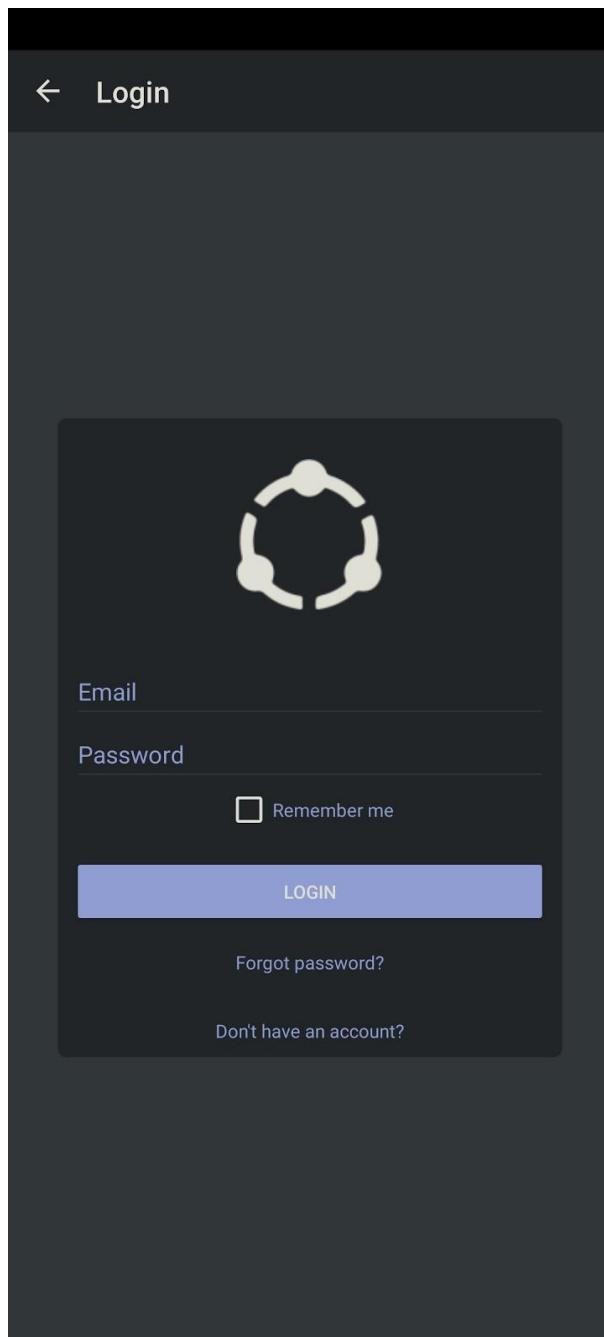


If you do not have an account yet, you can still use some functionalities of Platon. You can search for users, upcoming events, and even workspaces that are currently being worked on or published. You can see the trending projects.



If you want to be a part of Platon, you may register by providing your name, surname, email and your job. To go to the register page, click register on the right top.

Further, you will be able to link your Google Scholar and ResearchGate accounts to fetch your projects. You have to give an email address in use, since a verification code will be sent to it. You have to verify your account to log in. You have to accept Terms & Conditions to register.



Platon does not require a username for the users. You may log in with providing your email address and password. To log in, click on the LOGIN button on the right top at the home page.

If you ever forget your password, you can reset it by clicking on a link that will be sent to your email address.

You will be directed to the home page after logging in.

Your Profile:

Welcome to PLATON
Discover research and connect with your scientific community.

Upcoming Events

ACM WiSec 2021
Jun 28, 2021 - Jul 1, 2021

Activity Stream

- Christophe Daussy uploaded a new file named 'paper.pdf' to the workspace 'Predicting Earthquake with DL'
- Christophe Daussy created a new folder named 'Data Analysis' in the workspace 'Predicting Earthquake with DL'
- Christophe Daussy created a new folder named 'Initial Model Analysis' in the workspace 'Predicting Earthquake with DL'
- Christophe Daussy deleted the file named 'IMG-20190609-WA0001.jpg' in the workspace 'Predicting Earthquake with DL'
- Christophe Daussy deleted the file named 'Black Heart Emoji On Apple Ios 11 3.jpg' in the workspace 'Predicting Earthquake with DL'
- can bolukbas commented and rated Halil Umut Özdemir

Trending Projects

- Predicting Earthquake with DL (Popular)
- Medical Information System (Popular)
- Discovering Bias in Datasets (Popular)
- Image Denoising (Popular)
- Spacing in Basketball (Popular)
- Automated Conduction for Symphony (Popular)
- Unique ID management (Popular)

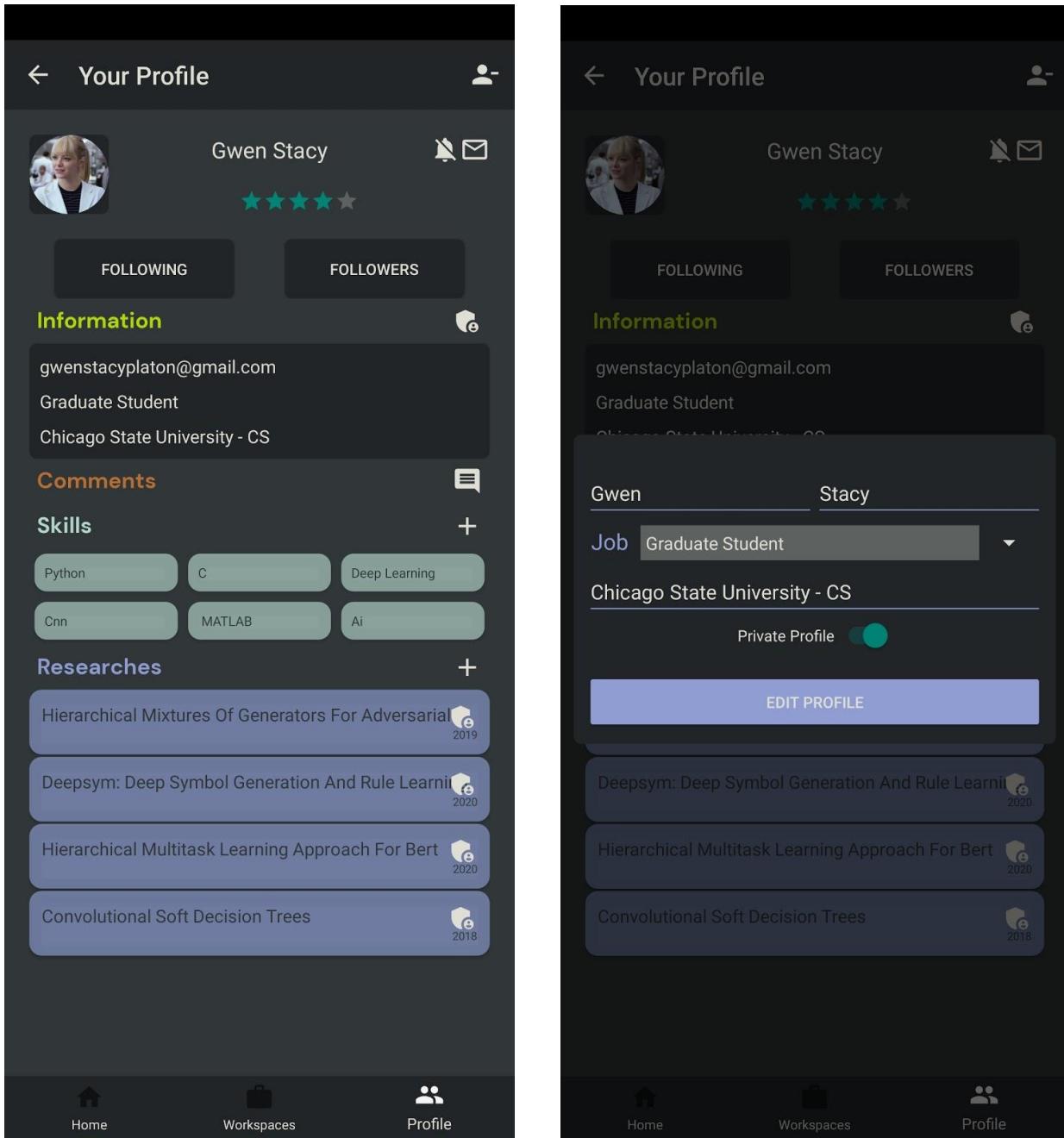
Trending Projects

Home Workspaces Profile

Home Workspaces Profile

To go to your profile, click on the profile icon on the right bottom.

You may personalize your profile by several ways:



If you are not satisfied with your current information, you may click the "Information" row to edit your profile.

Here, you change your current information. You can change the privacy of your profile (see Engaging With Other Profiles), update your institution, update your job and name.

The image consists of two side-by-side screenshots of a mobile application interface.

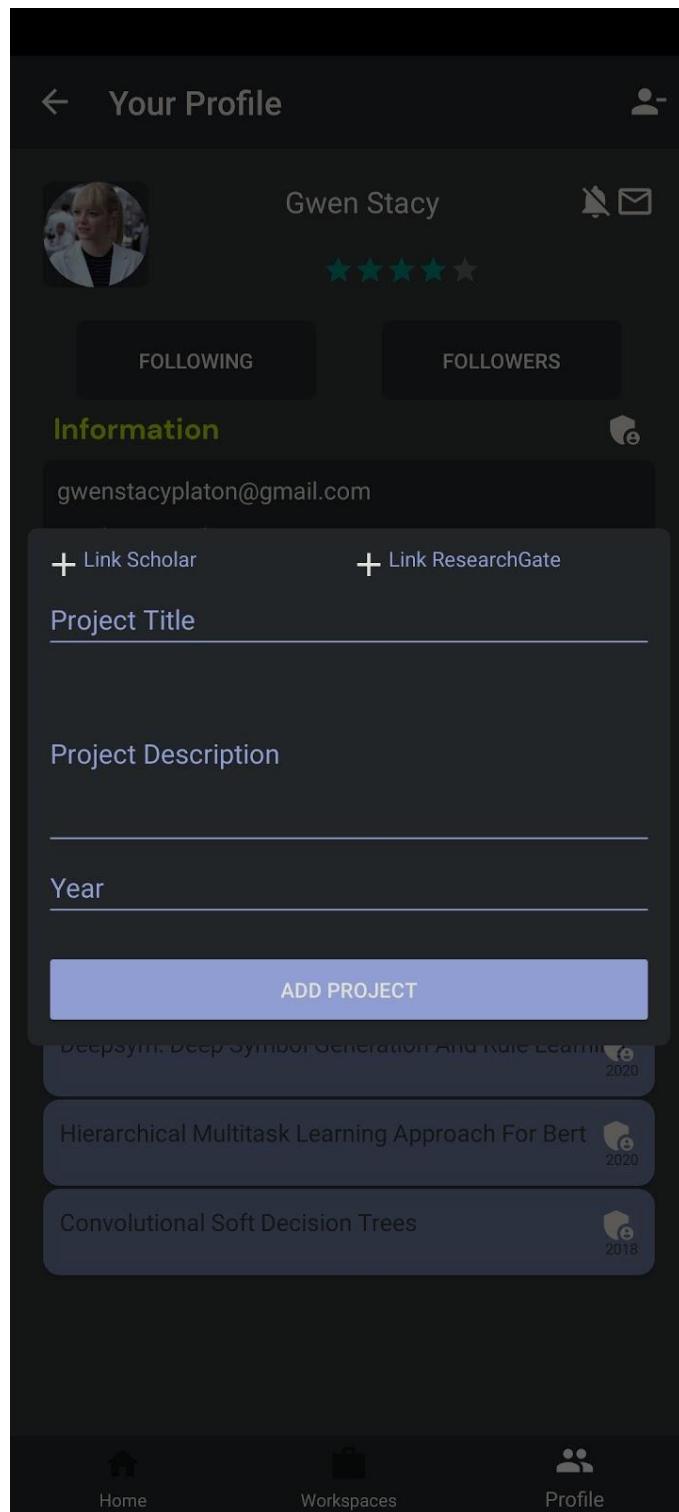
Left Screenshot: A modal window titled "Skills" is displayed over a dark background. It lists several skill names, each preceded by a checkbox. Some checkboxes are checked (e.g., MATLAB, Ai, C, Cnn, Deep Learning), while others are not (e.g., Deep Learning, Artificial Intelligence, Assembly, C++, Calculus, Cloud Computing, Context In Music, Control Theory, Data Science, Flask, Information Theory). At the bottom of the modal, there are two buttons: "ADD NONEXISTENT SKILL" and "COMPLETED".

Right Screenshot: The application's main profile screen for a user named "Gwen Stacy". The profile includes a circular profile picture, the name "Gwen Stacy", a 5-star rating, and buttons for "FOLLOWING" and "FOLLOWERS". Below this, the "Information" section displays the email "gwenstacyplaton@gmail.com", the title "Graduate Student", and the university "Chicago State University - CS". A large, semi-transparent overlay box is positioned over the bottom half of the screen. This overlay has a header "Skill" and a "Skill" input field. Below the input field is a grey button labeled "ADD". To the right of the "ADD" button, the word "COMPLETED" is written in green. Underneath this, a list of five completed projects is shown in cards:

- Hierarchical Mixtures Of Generators For Adversarial [2019]
- Deepsym: Deep Symbol Generation And Rule Learnin [2020]
- Hierarchical Multitask Learning Approach For Bert [2020]
- Convolutional Soft Decision Trees [2018]

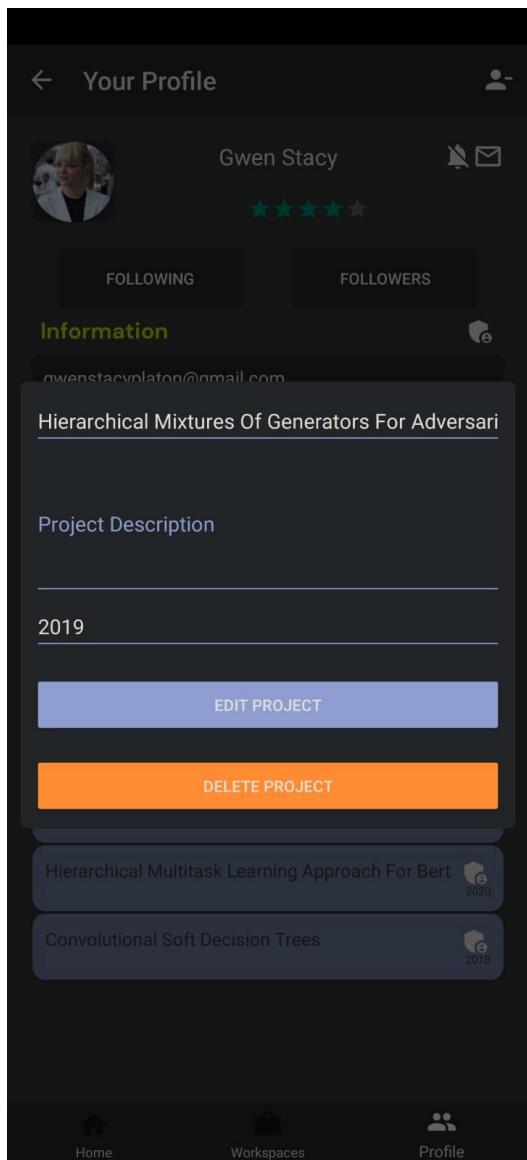
At the very bottom of the screen, there is a navigation bar with three icons: Home, Workspaces, and Profile.

You may add some of your skills. To do that, click on the plus button near the Skills section. You may add yourself an existing skill, or you may create a skill. You can choose from the list for existing skills, or just type the skill name you want to add and then click the add button.



You may also click the plus button near the Research section to add your past research. You need to provide your project's title, a description of it, and also a date. Date and title are mandatory.

You can link your Google Scholar and ResearchGate accounts. If you link your account, the projects you registered there will be automatically uploaded to your profile.



You can edit your research information later on by clicking the button near the research you want to update.

You can also delete it by clicking the delete button, if you change your mind.



Gwen Stacy

5 ★★★★☆

FOLLOWING **FOLLOWERS**

Information

gwenstacyplaton@gmail.com
Graduate Student
Chicago State University - CS

Comments

Skills

Python C Deep Learning
Cnn MATLAB Ai

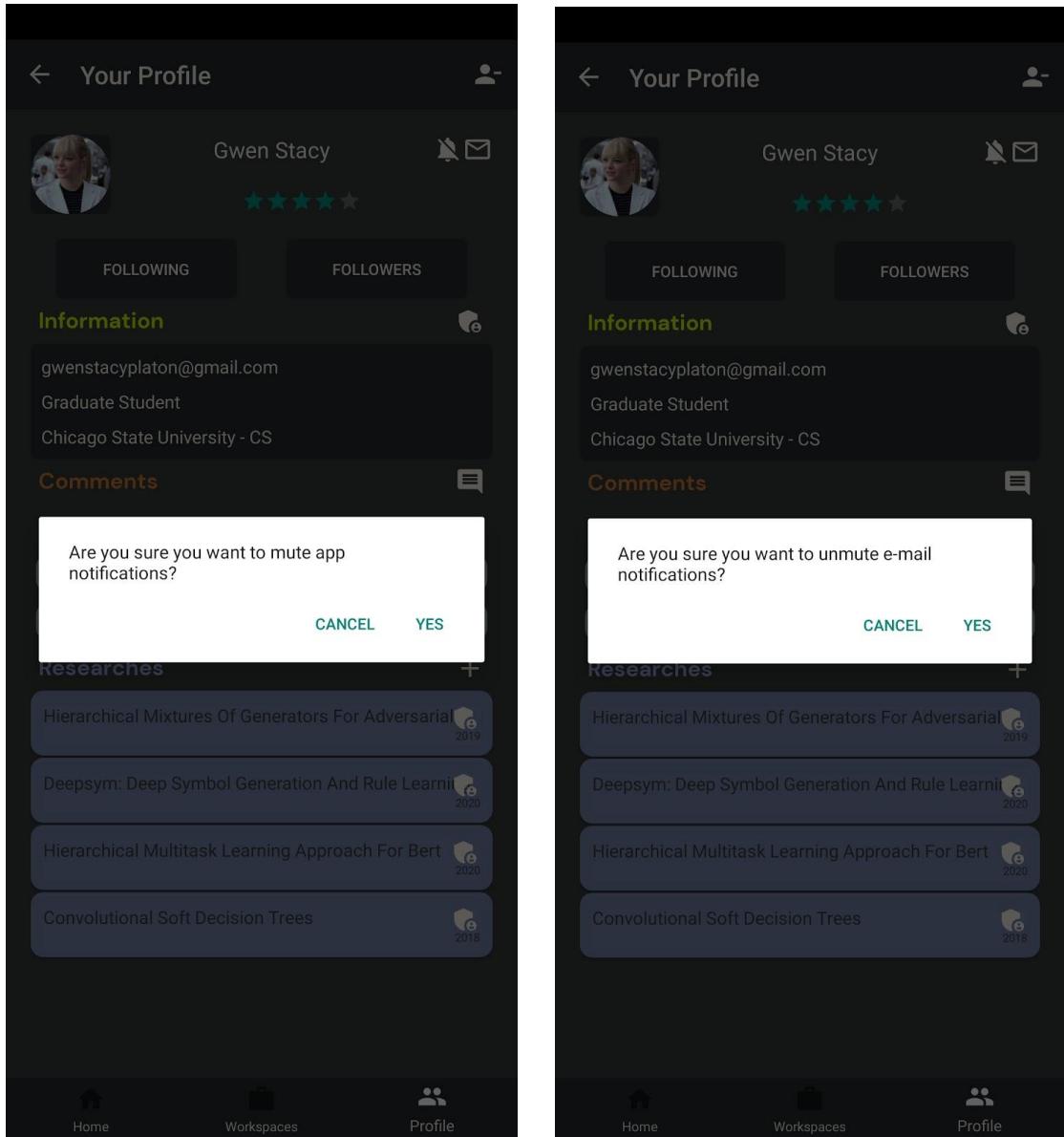
Researches

Hierarchical Mixtures Of Generators For Adversarial [E] 2019
Deepsym: Deep Symbol Generation And Rule Learning [E] 2020
Hierarchical Multitask Learning Approach For Bert [E] 2020
Convolutional Soft Decision Trees [E] 2018

Home **Workspaces** **Profile**

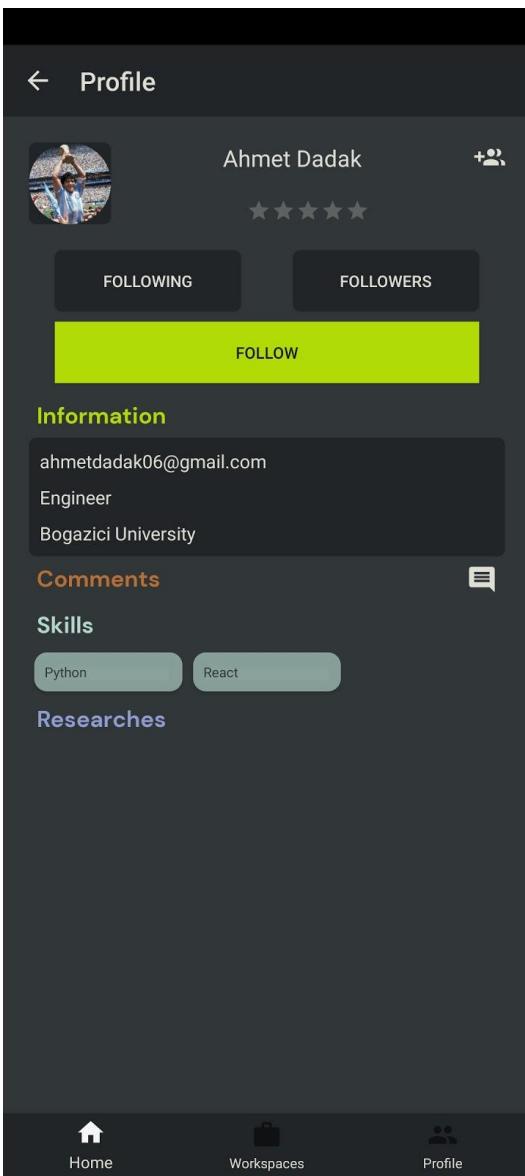
All Photos

Thumbnail	Title	Details
	Predicting Earthquake with DL	Project
	Medical Information System	Project
	Discovering Bias in Datasets	Project
	Image Denoising	Project
	Scoring in Basketball	Project
	Automated Conduction for Symphony	Project
	...	Project
	Observing Black Holes and the name Black Hole (End On)	Post
	... comment and about their time	Post
	...	Post
<img alt="Thumbnail 381		

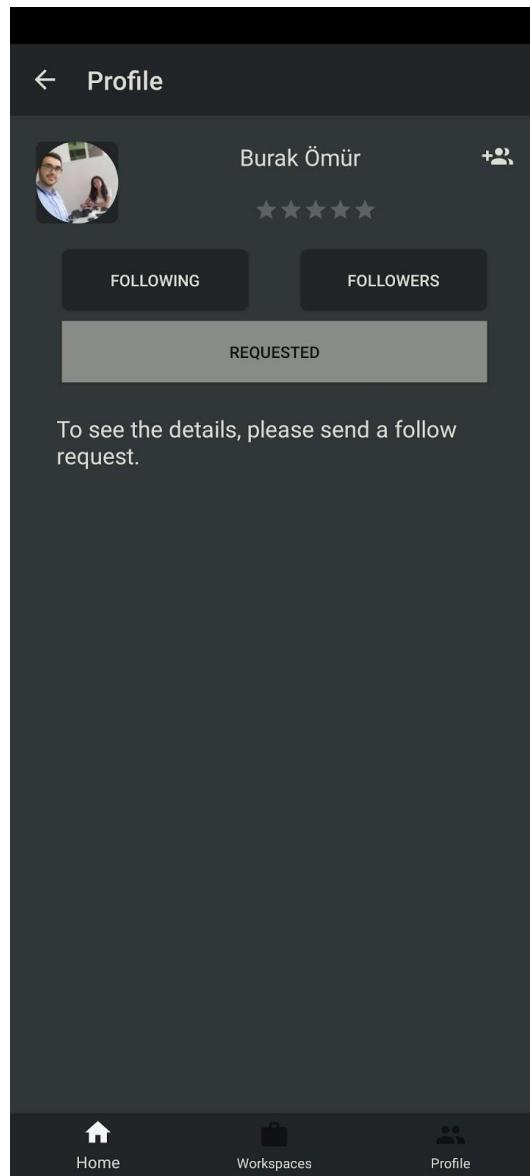


You can mute/unmute your notifications as for both mail and in-app using the buttons besides the name of the person.

Engaging with Other Profiles:



Public Profile



Private Profile

There are two types of profiles in Platon: public and private. If you choose the privacy of your account as public, other profiles can see your name, surname, job, institution, skills, followers, following, and past research information. Registered users will be able to follow you without sending a follow request first.

If you want a private profile, other users and guests can only view your name, surname, profile photo and your job. To view your other personal information, they will have to send you a follow request. You will have the option to accept and reject their request.

If you accept, they will view your information as if you had a public account.

Your Profile

Gwen Stacy

FOLLOWING FOLLOWERS

Information

gwenstacyplaton@gmail.com
Graduate Student
Chicago State University - CS

Comments

Skills

Python C Deep Learning
Cnn MATLAB Ai

Researches

Hierarchical Mixtures Of Generators For Adversarial 2019
Deepsym: Deep Symbol Generation And Rule Learnin 2020
Hierarchical Multitask Learning Approach For Bert 2020
Convolutional Soft Decision Trees 2018

Home Workspaces Profile

Details

can bolukbas
Oyku Yilmaz
Halil Umut Özdemir
Ertugrul Bulbul
Christophe Daussy

Home Workspaces Profile

You can see followers and following in your profiles under the tabs named "FOLLOWERS", "FOLLOWING", respectively.

The image consists of two side-by-side screenshots of a mobile application interface, likely from the Platon app. Both screenshots show a dark-themed interface with a navigation bar at the bottom containing three icons: a house (Home), a briefcase (Workspaces), and a people icon (Profile).

Screenshot 1 (Left): Details Screen

This screen shows a list of users under the heading "Details". Each item includes a small profile picture, the user's name, and a plus sign icon for adding them.

- can bolukbas
- Oyku Yilmaz
- Halil Umut Özdemir
- Ertugrul Bulbul
- Christophe Daussy

Screenshot 2 (Right): Details Screen with Recommendations

This screen also shows a list of users under the heading "Details". A "Recommended" section is overlaid on the list, containing a grid of user profiles and names.

Recommended	
Ahmet Eker	
Berkay Karaman	
Sena Ergisi	
Burhan Can Akkus	
Ahmet Dadak	
Cagri Ciftci	
Umutcan Uvut	
Osman Alp Toprak	
Bilge Ozdemir	
Alperen Divriklioglu	

Platon offers a recommendation system. You can see your recommendations for following users on the right top of the following or followers section. The suggestions will be according to the match rate of your and other users' skills.

Other Profiles

The image displays two side-by-side screenshots of a mobile application interface, likely a professional networking or collaboration tool.

Screenshot 1 (Left): Profile Overview

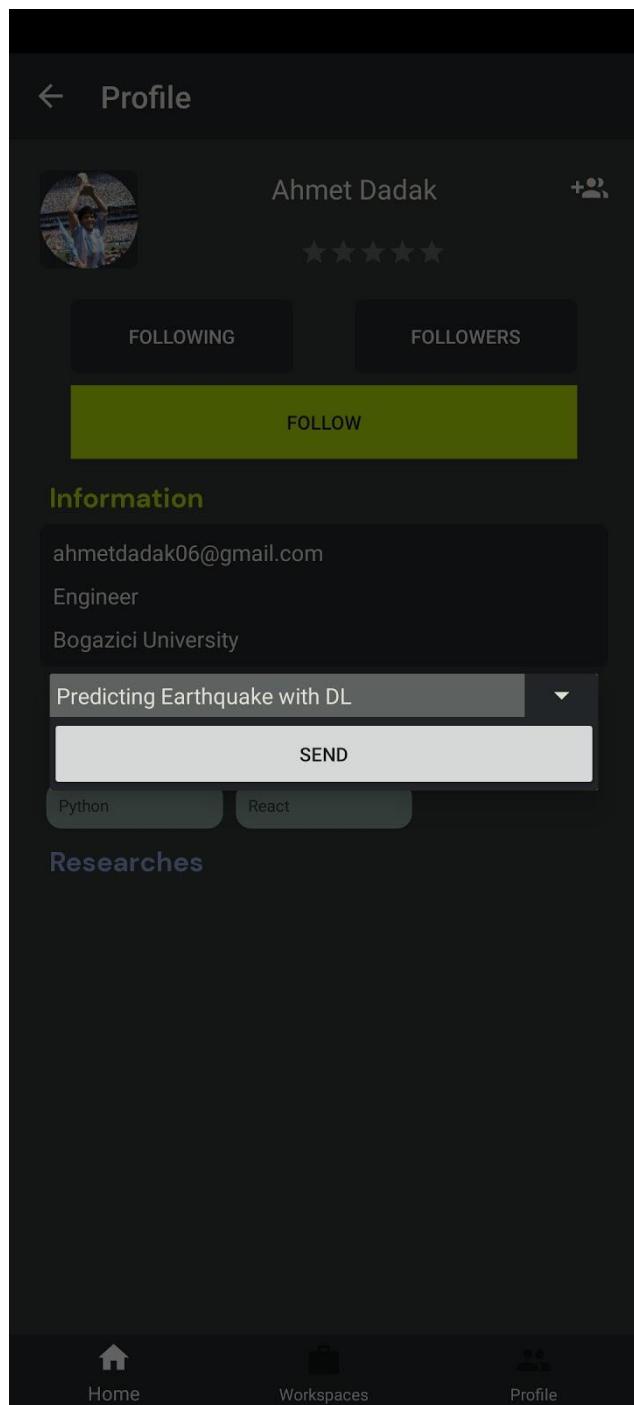
- User Info:** Christophe Daussy, 5 stars.
- Following/Followers:** Following (button), Followers (button).
- Information:** christophe@gmail.com, Professor, Chicago State University.
- Comments:** Comment: 0, Really a hard working person, posted on Wed, 27 Jan 2021 16:00:24 GMT.
- Skills:** Python, C++, Deep Learning.
- Researches:** Hierarchical Mixtures Of Generators For Adversarial (2019), Deepsym: Deep Symbol Generation And Rule Learning (2020), Hierarchical Multitask Learning Approach For Bert (2020), Convolutional Soft Decision Trees (2018).

Screenshot 2 (Right): Comment Interaction

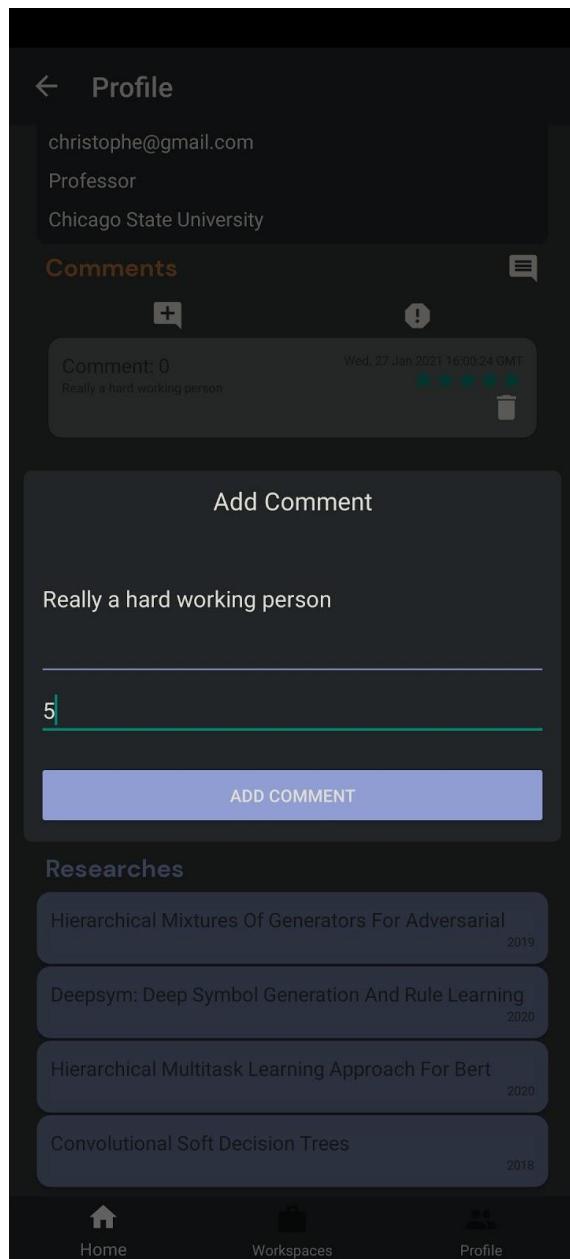
- User Info:** christophe@gmail.com, Professor, Chicago State University.
- Comments:** Comment: 0, Really a hard working person, posted on Wed, 27 Jan 2021 16:00:24 GMT.
- Add Comment:** Input field contains "Really a hard working person", a rating scale shows "5", and a blue "ADD COMMENT" button.
- Researches:** Hierarchical Mixtures Of Generators For Adversarial (2019), Deepsym: Deep Symbol Generation And Rule Learning (2020), Hierarchical Multitask Learning Approach For Bert (2020), Convolutional Soft Decision Trees (2018).

Bottom Navigation: Home, Workspaces, Profile.

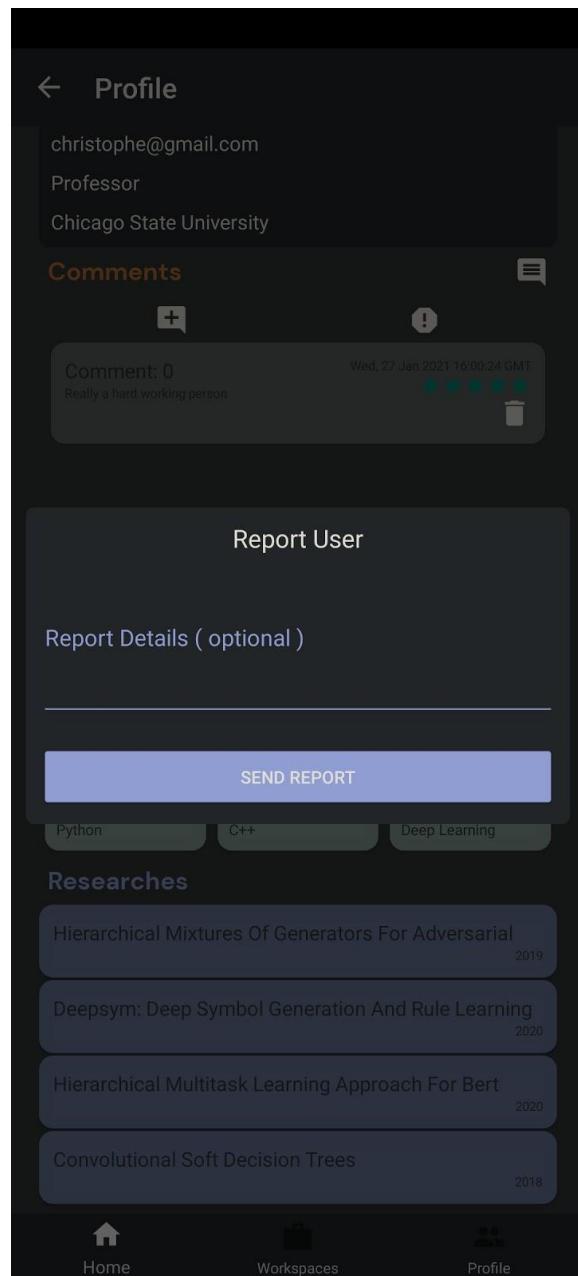
Other users you have collaborated with before may leave comments and rate you. You cannot alter these comments, or delete them. To see your comments, go to the comments section on your profile.



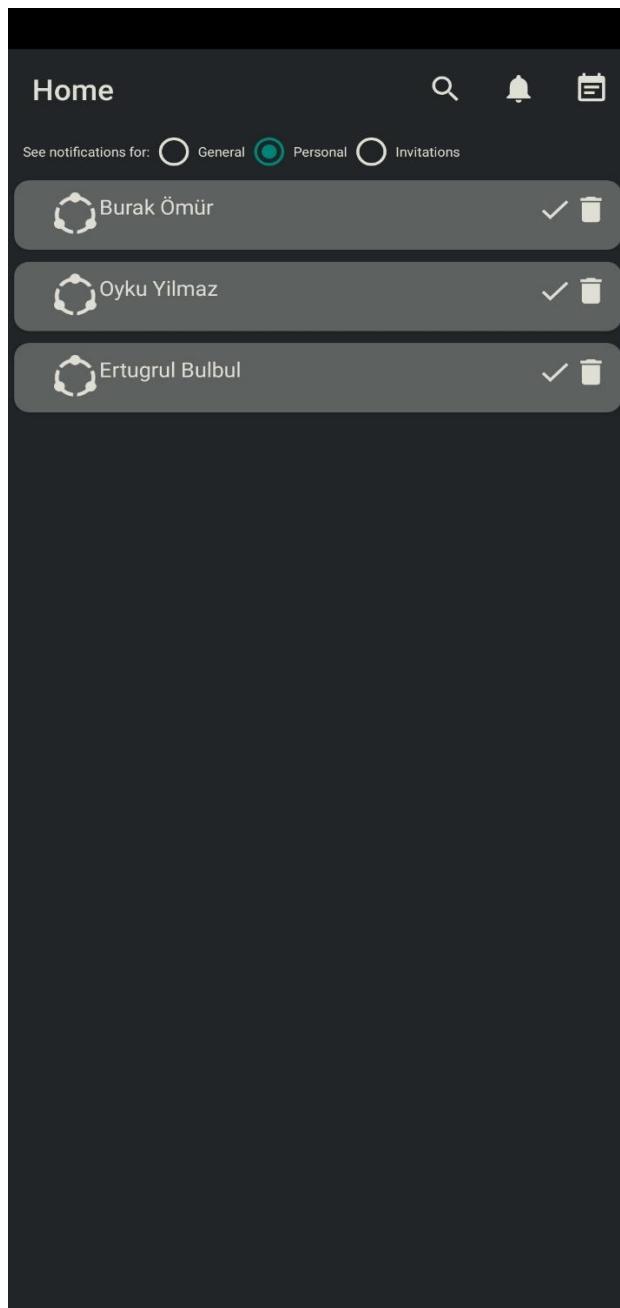
Collaborators of workspaces can invite other people to collaborate with them. To do that, you need to click the add button near the user's name.



Other users you have collaborated with before may leave comments and rate you. You cannot alter these comments, or delete them. To see your comments, go to the comments section on your profile.

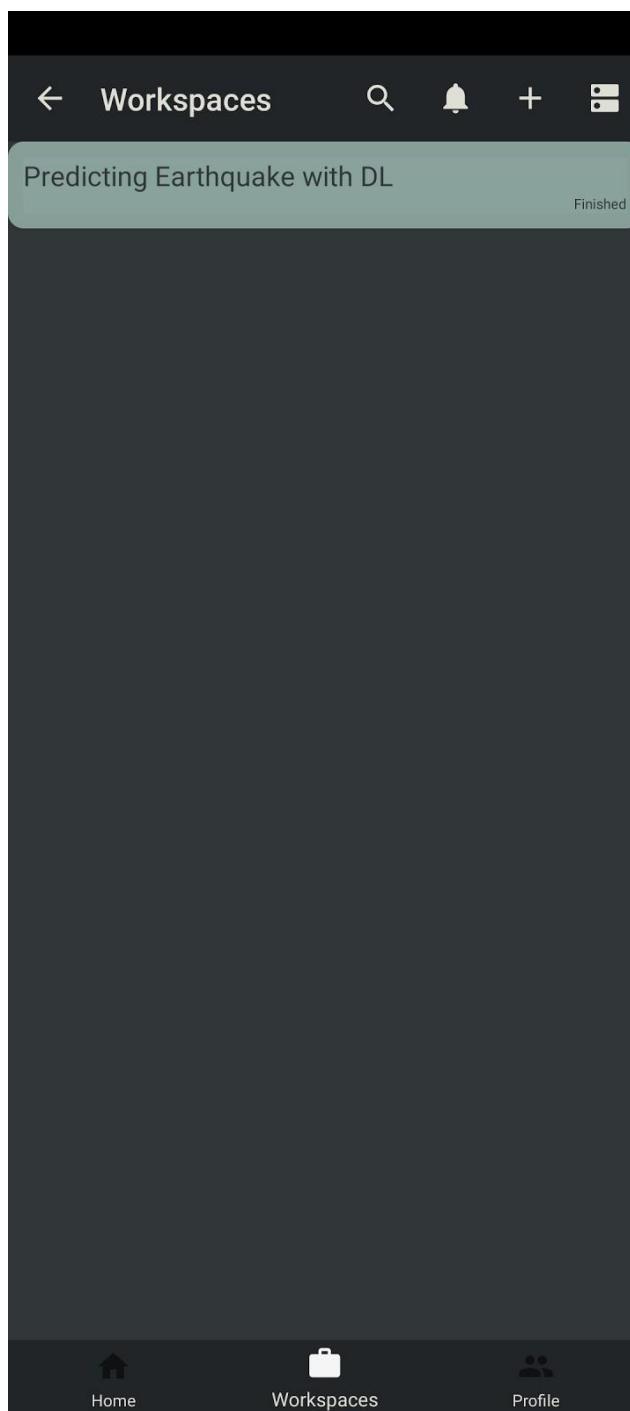


You can report with defected users to administrators of Platon by clicking the report button and filling the form.

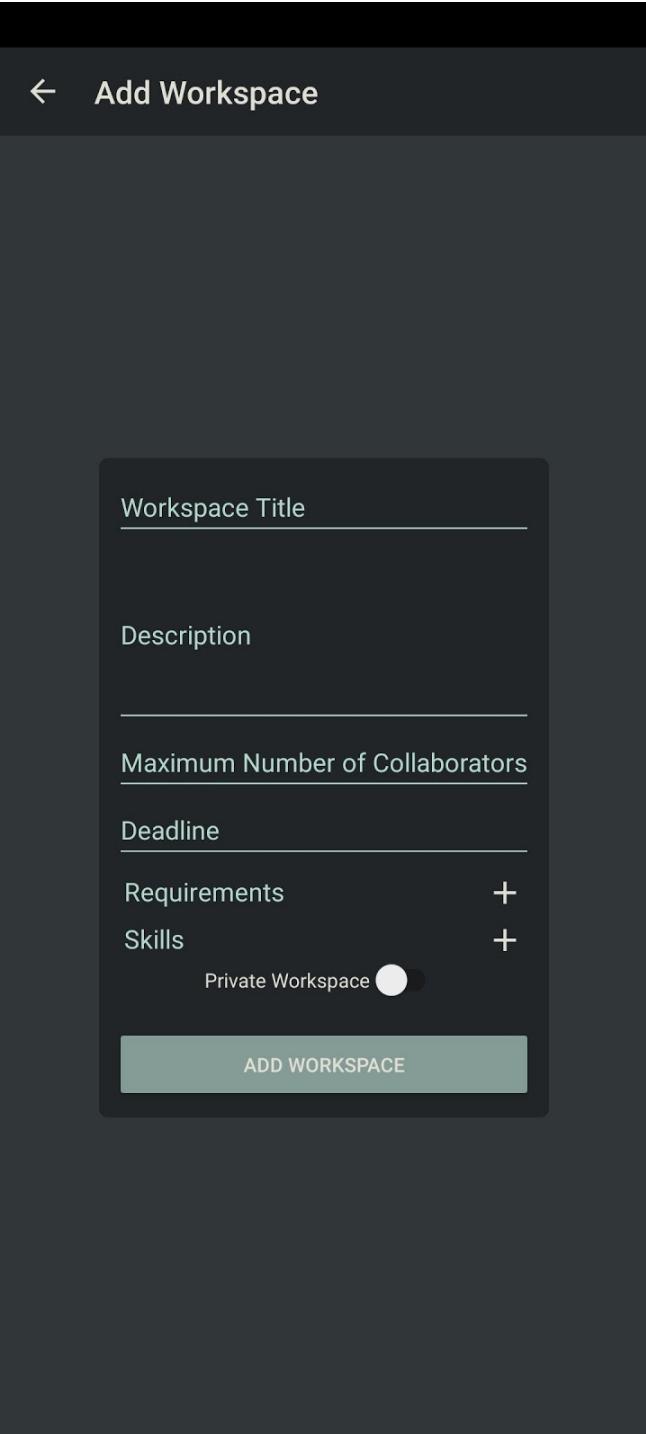


If you have a private account, users cannot directly follow you, you need to accept their requests first. To answer the following requests, go to the PERSONAL at the notifications section.

Workspaces:



Workspaces are storage areas for a group of users to develop their project/paper proposal collaboratively. To list your workspaces, click on the workspace icon on the bottom. If you do not have a workspace yet, you can create a workspace by clicking the add button. To create a workspace, you must provide a title and a description associated with the workspace.



The workspace is created as public by default. Public workspace means any user or guest may view the details of your workspace, files, issues, milestones, and the contributors in your workspace. They may search for it from the search bar, and your workspace may be seen in the trending projects, if it gets much attention. After viewing, a user may send a collaboration request to join your workspace.

If you want your workspace to be invisible, you should make it private. Private workspaces do not show up in any search or in the trending projects section. Only the collaborators may view it.

You can provide the maximum number of collaborators in your workspace. The default number is 10 if you decide to leave it empty.

You may write the requirements to join the workspace under the Requirements section.

You may add the required skills in your workspace. Skills will act as labels to your workspace. If you set your workspace to be public, other users may use those skills to narrow down their search. For example, if your workspace has a skill named "Python", then if the user filters their search by giving python as a parameter, your workspace will be visible to their search result.

You may provide a deadline to your workspace.

To inspect a workspace, click on the workspace.

Here you will see the details of your workspace.

Tags on the detail section are clickable. By clicking on them you can see the workspaces that have the same tag.

The screenshot shows a detailed view of a workspace titled "Predicting Earthquake with DL".

Information: By using some of the deep learning methods, it is possible to predict earthquakes. Our aim is to predict earthquakes to decrease the loss of life.

Description: Public Workspace, Maximum Number of Collaborators: 2, Due 2021.01.30, Finished.

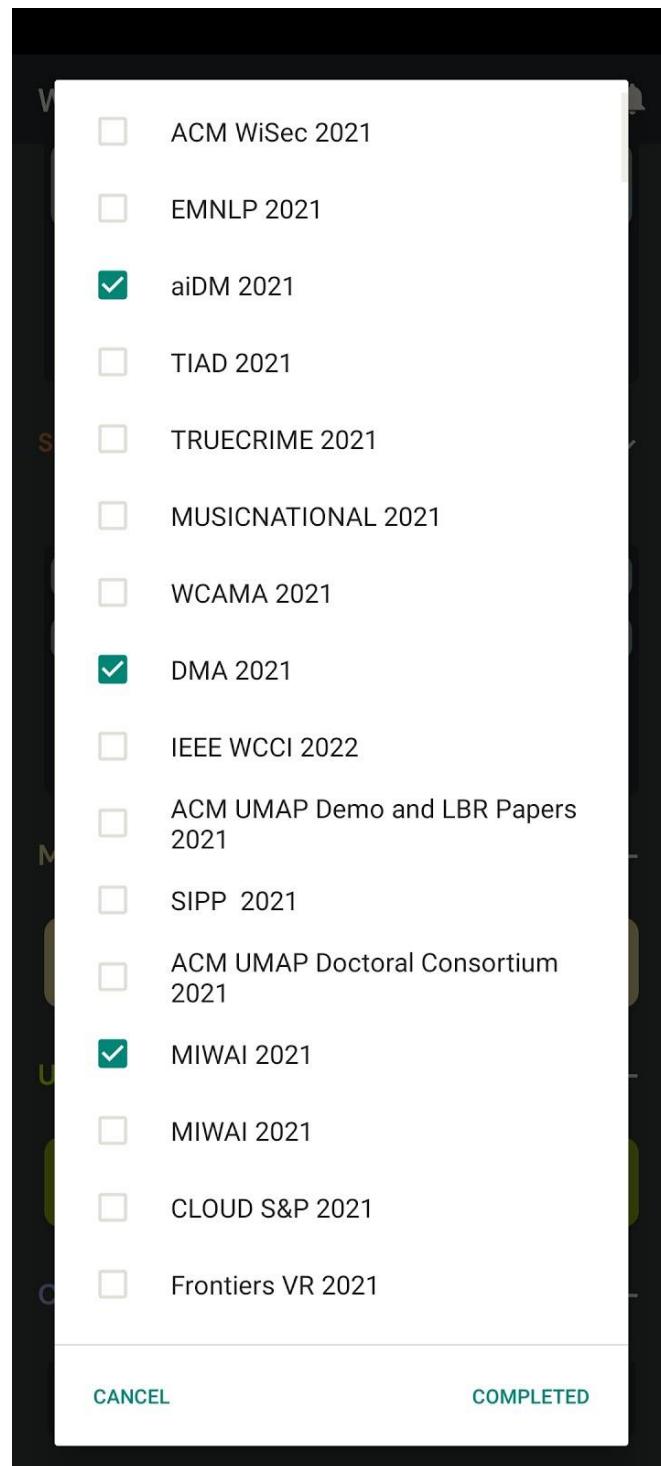
Requirements: - Knowledge of Deep Learning
- Knowledge of Data Science
- Knowledge of Signal Processing

Skills: Python, C, Deep Learning, Machine Learning, MATLAB, Artificial Intelligence

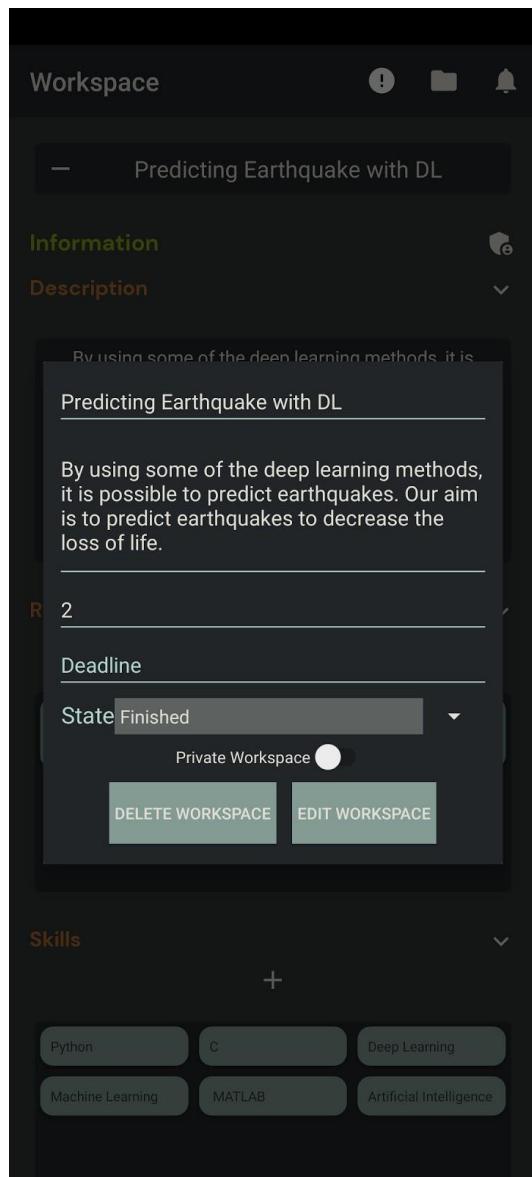
Milestone: Examine Dataset (Thu, 10 Dec 2020 23:59:00 GMT)

Upcoming Events: aiDM 2021 (Jun 5, 2021 - Jun 5, 2021)

Collaborators: Gwen Stacy, Christophe Daussy



You may link one or several upcoming events to your workspace. That will mean the project will also be a part of the specified upcoming event/s. Click on the upcoming events to do that.



To edit your workspace you need to click the edit button.

A workspace has three states:

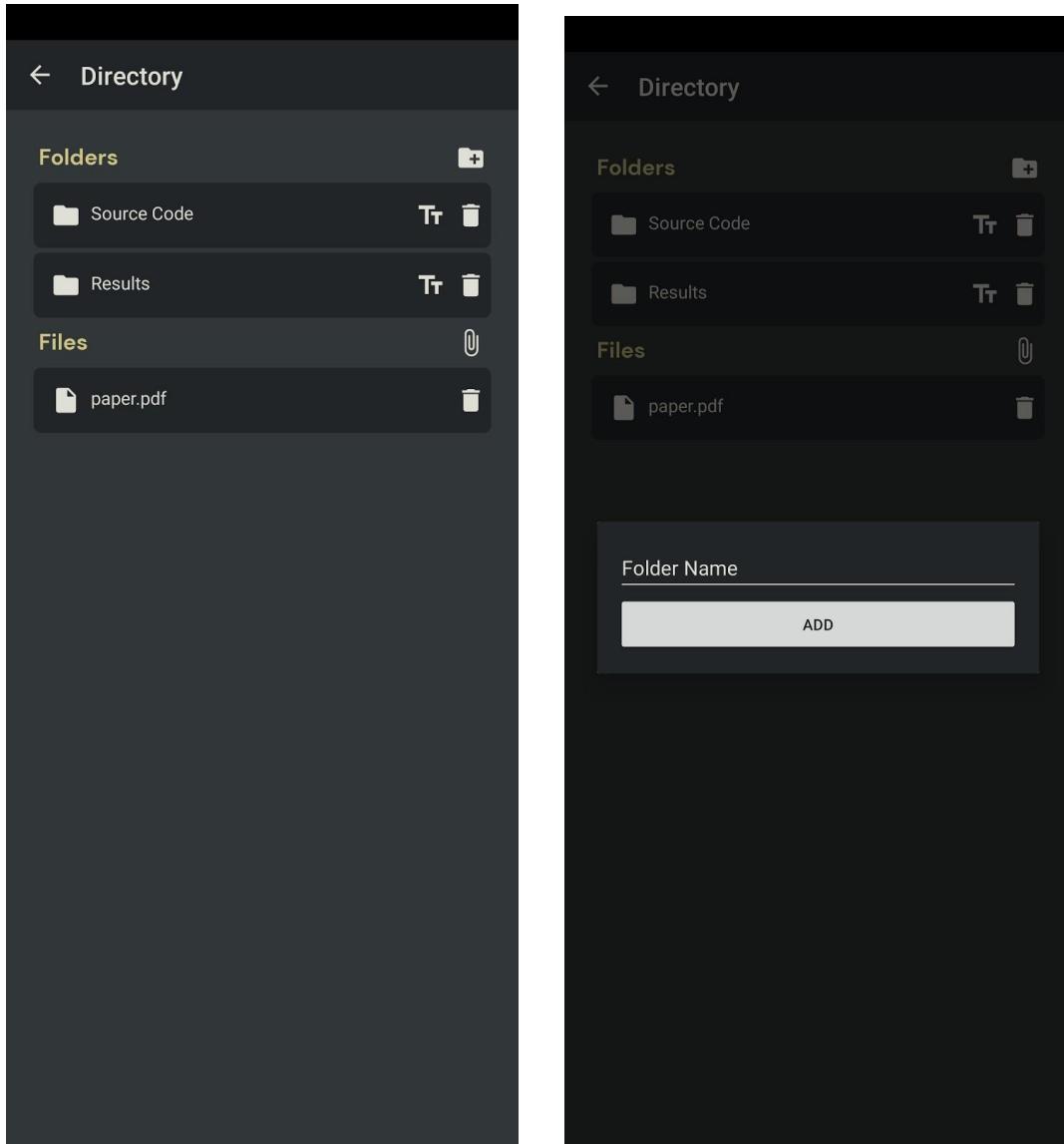
1- Search for Collaborators State: It is the first of the three states of a project. In this state, the founder of the project waits for requests from other Users and sends invitations to Users. A workspace is created with this state by default.

2- Ongoing State: It is the second of the three states of a project. This is the development phase of the project. After finding enough collaborators, the project will go to the second state which is the Ongoing State.

3- Published State: It is the last stage of a project. After milestones are completed with the request of owners of the project, the project will go to Published State.

You may change the state of the project by editing your workspace.

Files



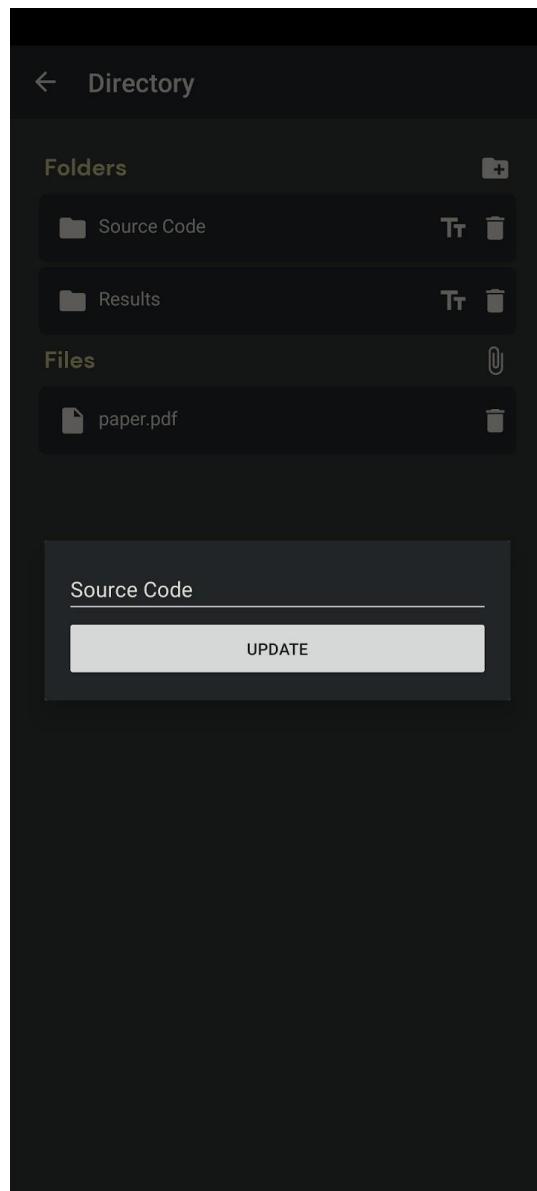
You may add files related to the workspace. You may also create folders for ease of use.

To add or view your files, go to the files section under the workspace page.

To create a folder, type the name of the folder to the text view on the top of the create folder button. Then click the ADD button.

To delete it, click the delete button on the right of the desired folder. This cannot be undone.

You may check inside a folder by clicking on the name of the folder.

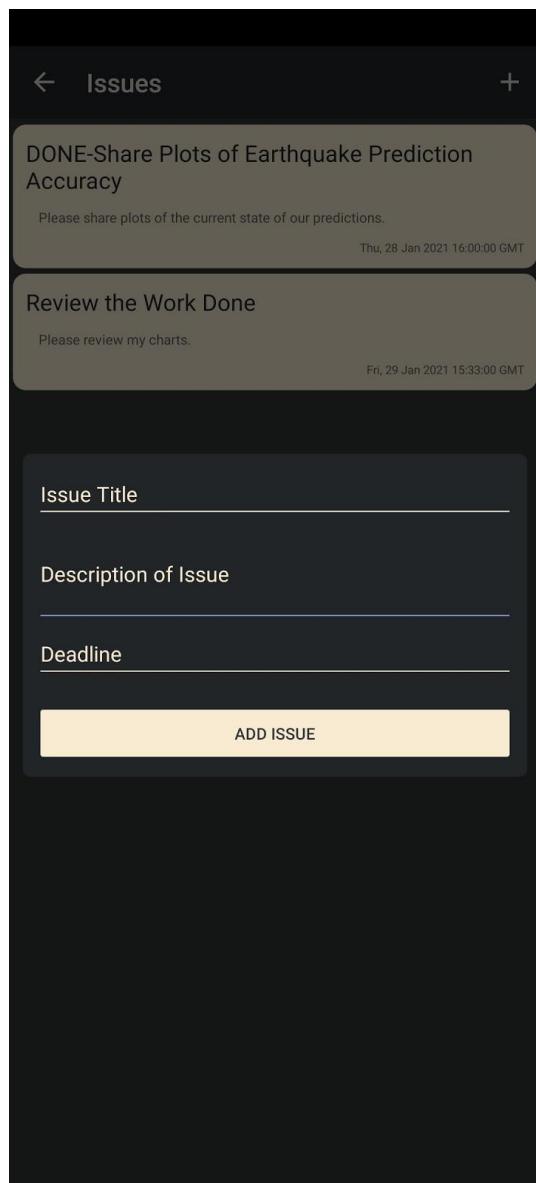
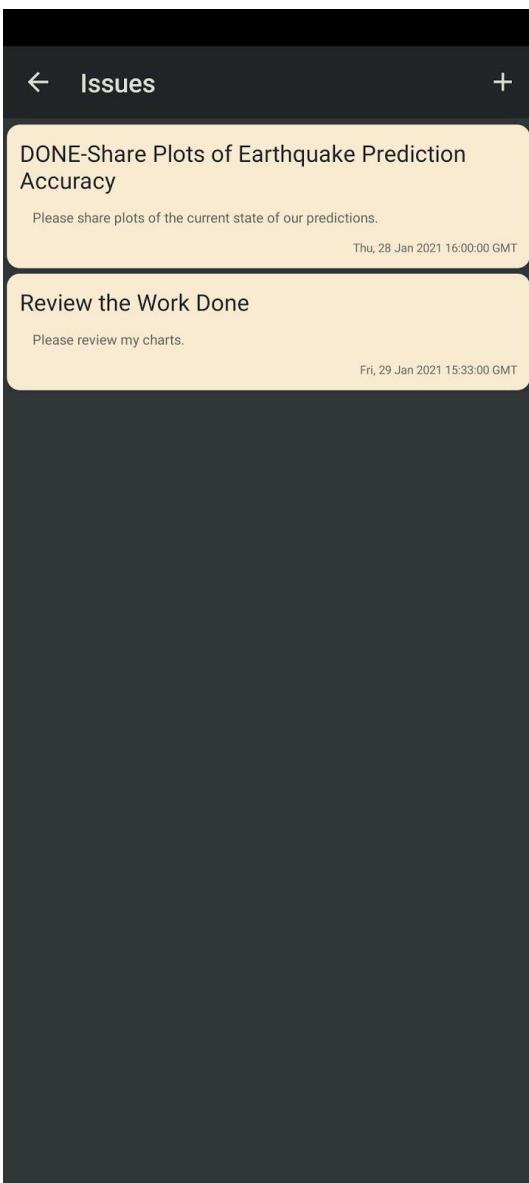


To rename a folder, click the rename button on the right of the desired folder.

To upload a file, you should click the Choose file button. After you choose your file, you may specify its name. Once you upload it, you will not be able to change its name.

To delete it , click the delete button on the right of the desired file. This cannot be undone.

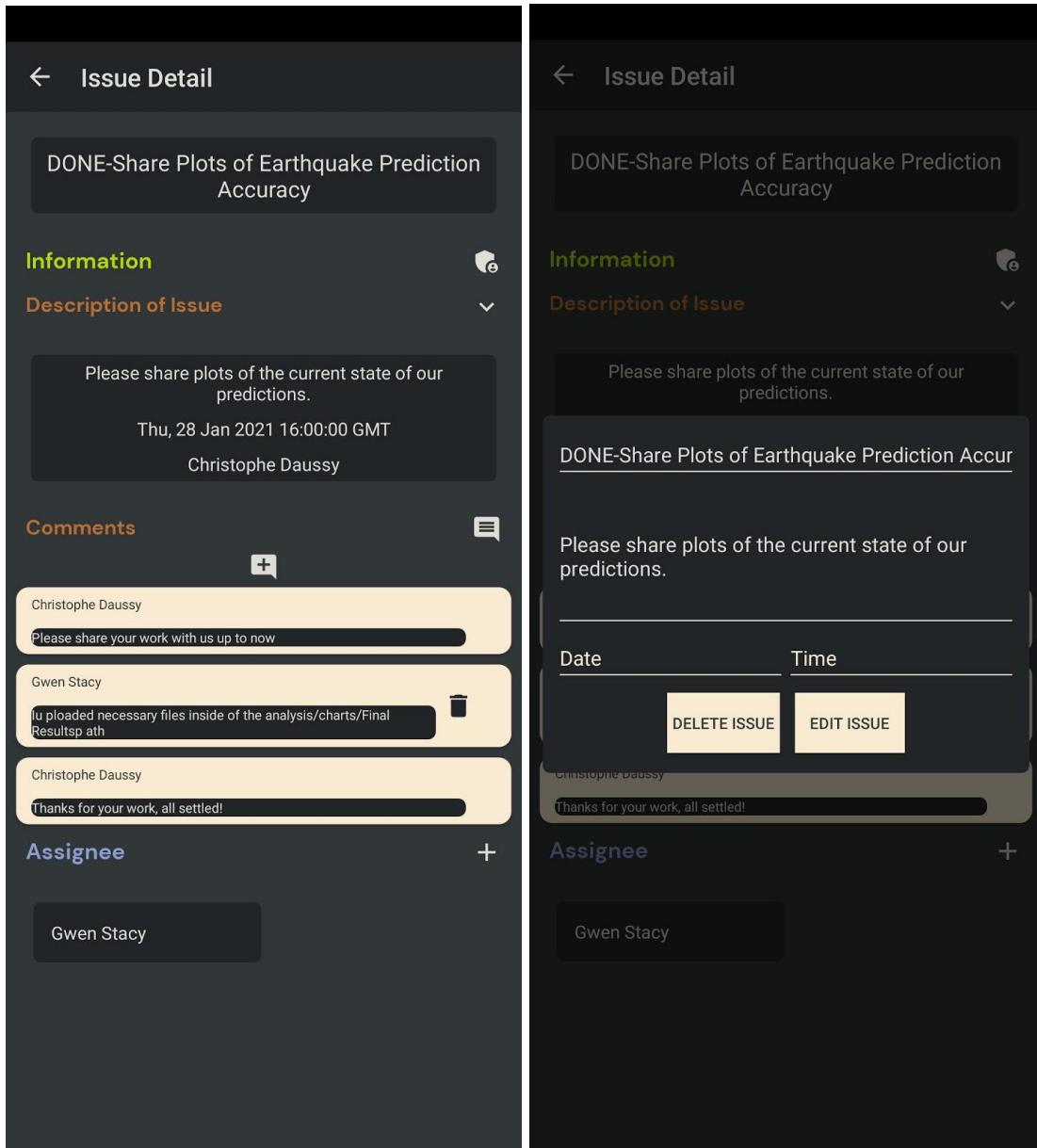
Issues:



You may create issues to keep track of your future work. To create an issue, go to the issues tab under your workspace's page.

To create an issue, click on the button.

You must provide a title, description and a deadline to the issue to be created.



To edit an issue, click on the edit button near that issue. You can also delete an issue. This cannot be undone.

You can also delete your previous comments by clicking the orange buttons near those comments.

The image displays two side-by-side screenshots of a mobile application's 'Issue Detail' screen. Both screens show a dark-themed interface with white text and light-colored buttons.

Left Screen (Initial State):

- Information:** DONE-Share Plots of Earthquake Prediction Accuracy
- Description of Issue:** Please share plots of the current state of our predictions.
- Comments:** A text input field labeled 'ADD COMMENT' and a message from Christophe Daussy: 'I uploaded necessary files inside of the analysis/charts/Final Results path'.
- Assignee:** Gwen Stacy is listed under 'Assignee'.

Right Screen (After Interaction):

- Information:** DONE-Share Plots of Earthquake Prediction Accuracy
- Description of Issue:** Please share plots of the current state of our predictions.
- Comments:** A text input field labeled 'ADD COMMENT' and a message from Christophe Daussy: 'I uploaded necessary files inside of the analysis/charts/Final Results path'.
- Assignee:** A modal dialog box titled 'Assignee' is open, listing two options:
 - Gwen Stacy (checkbox checked)
 - Christophe Daussy (checkbox unselected)A button labeled 'ADD ASSIGNEE' is visible at the bottom right of the dialog.

Issues may have comments. You can add comments by clicking.

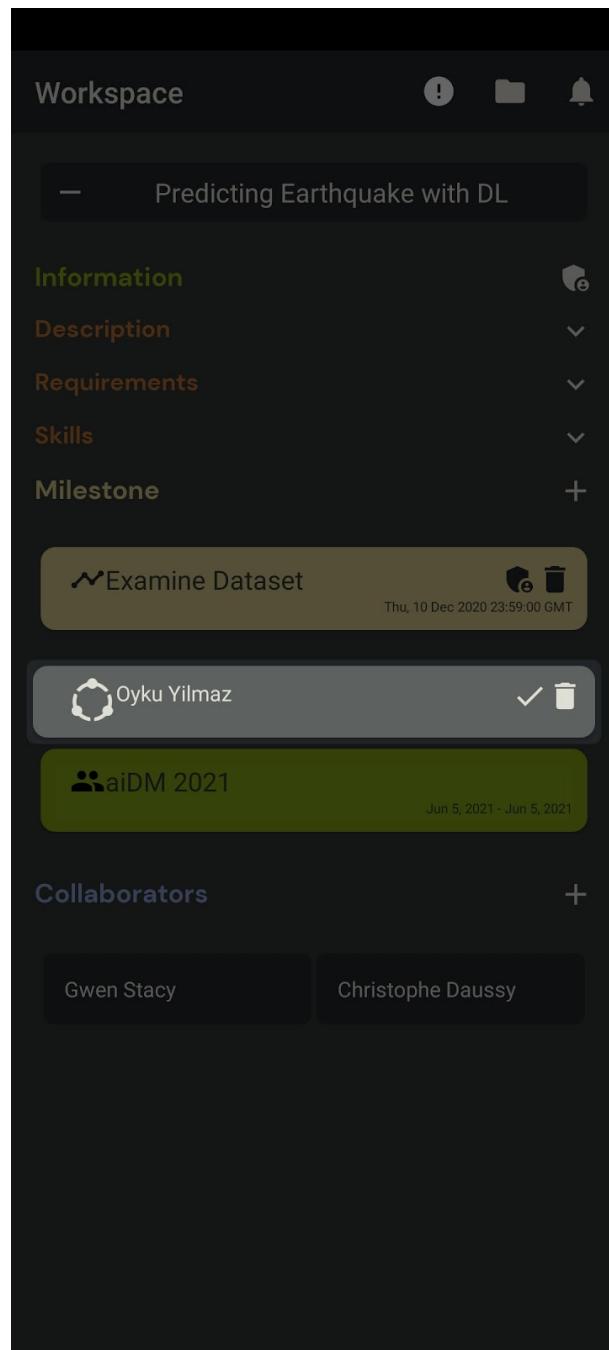
Milestones:

The image displays two side-by-side screenshots of a mobile application's workspace interface. Both screens show a header with 'Workspace' and three icons: a red exclamation mark, a folder, and a bell. Below the header is a dark grey box containing a bulleted list of skills: '- Knowledge of Deep Learning', '- Knowledge of Data Science', and '- Knowledge of Signal Processing'. The main area is titled 'Skills' with a dropdown arrow. A large yellow '+' button is positioned above a form for creating a milestone. The left screenshot shows a 'Milestone Title' field with 'Examine Dataset' typed into it, a 'Description' field with 'Examine Dataset of Earthquake Data', a 'Due' section with 'Date' and 'Time' fields, and a yellow 'ADD MILESTONE' button. The right screenshot shows the same fields after being filled out, with the 'ADD MILESTONE' button replaced by a yellow 'UPDATE' button. Below these forms are sections for 'Upcoming Events' and 'Collaborators'. The 'Upcoming Events' section shows an event for 'aiDM 2021' from Jun 5, 2021, to Jun 5, 2021. The 'Collaborators' section lists 'Gwen Stacy' and 'Christophe Daussy'.

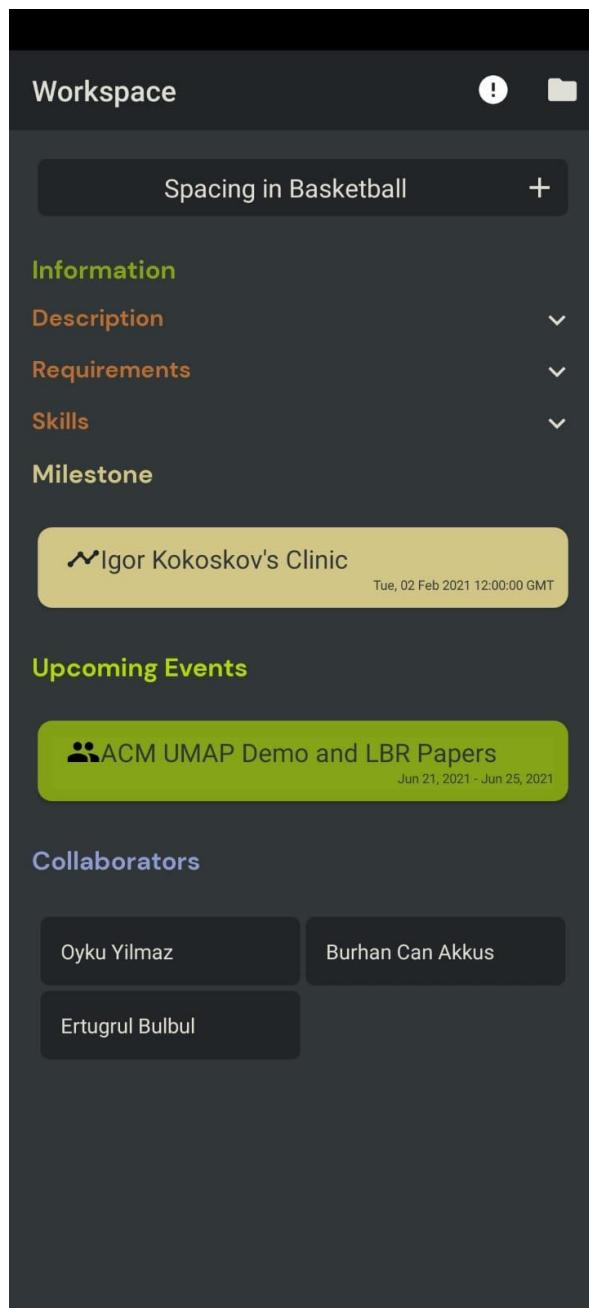
You may create milestones to your workspace. To create a milestone, go to the Milestones tab under your workspace's page.

After creating the milestone, you may edit its title, description and the deadline.

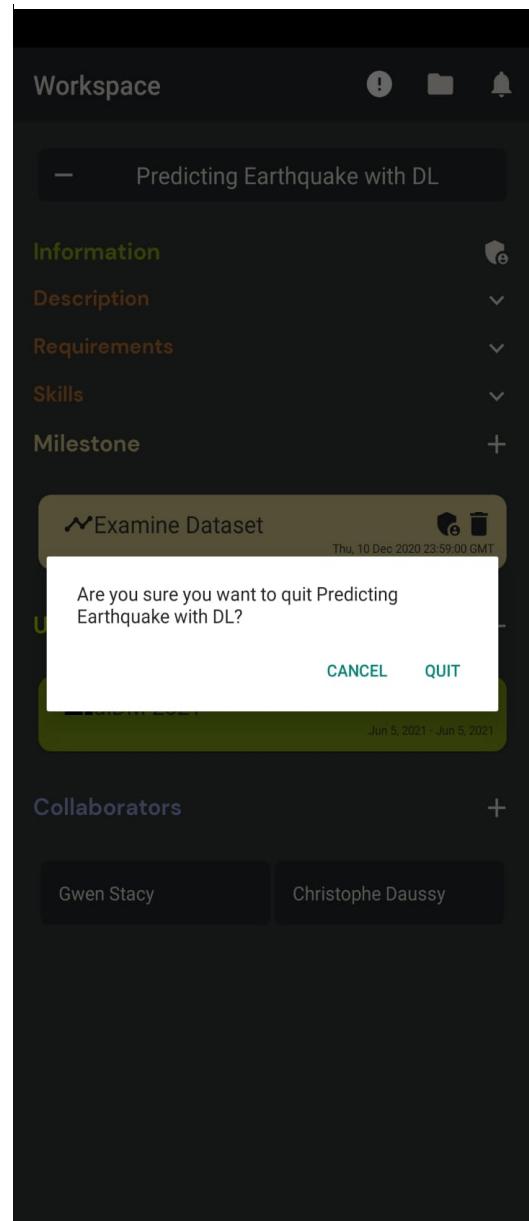
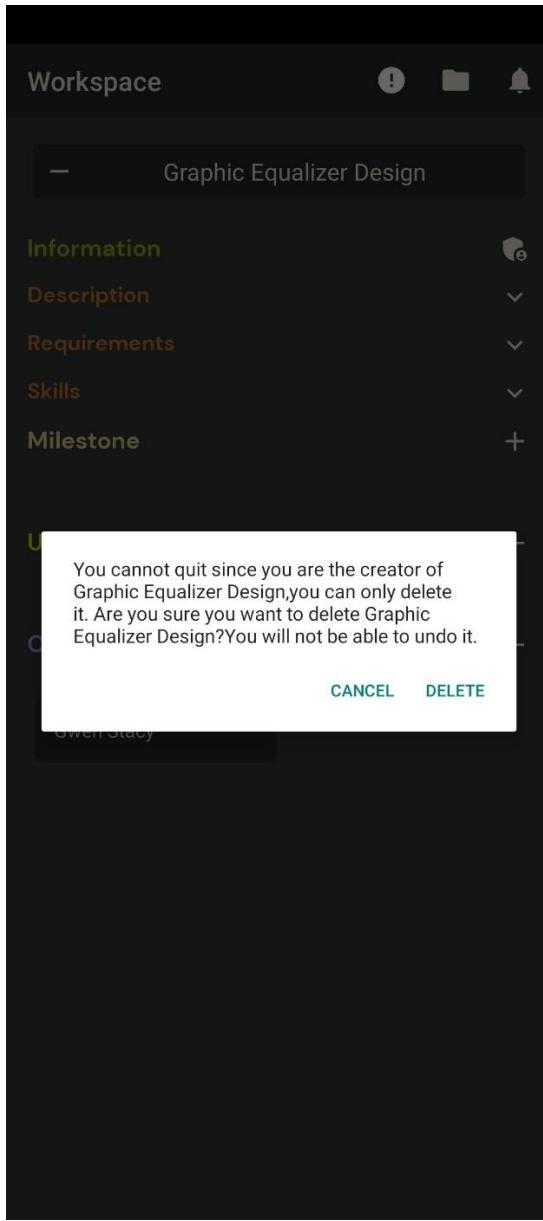
To delete a milestone, click on the delete button on the right of the milestone. This cannot be undone.



Here you may answer your workspace applications.

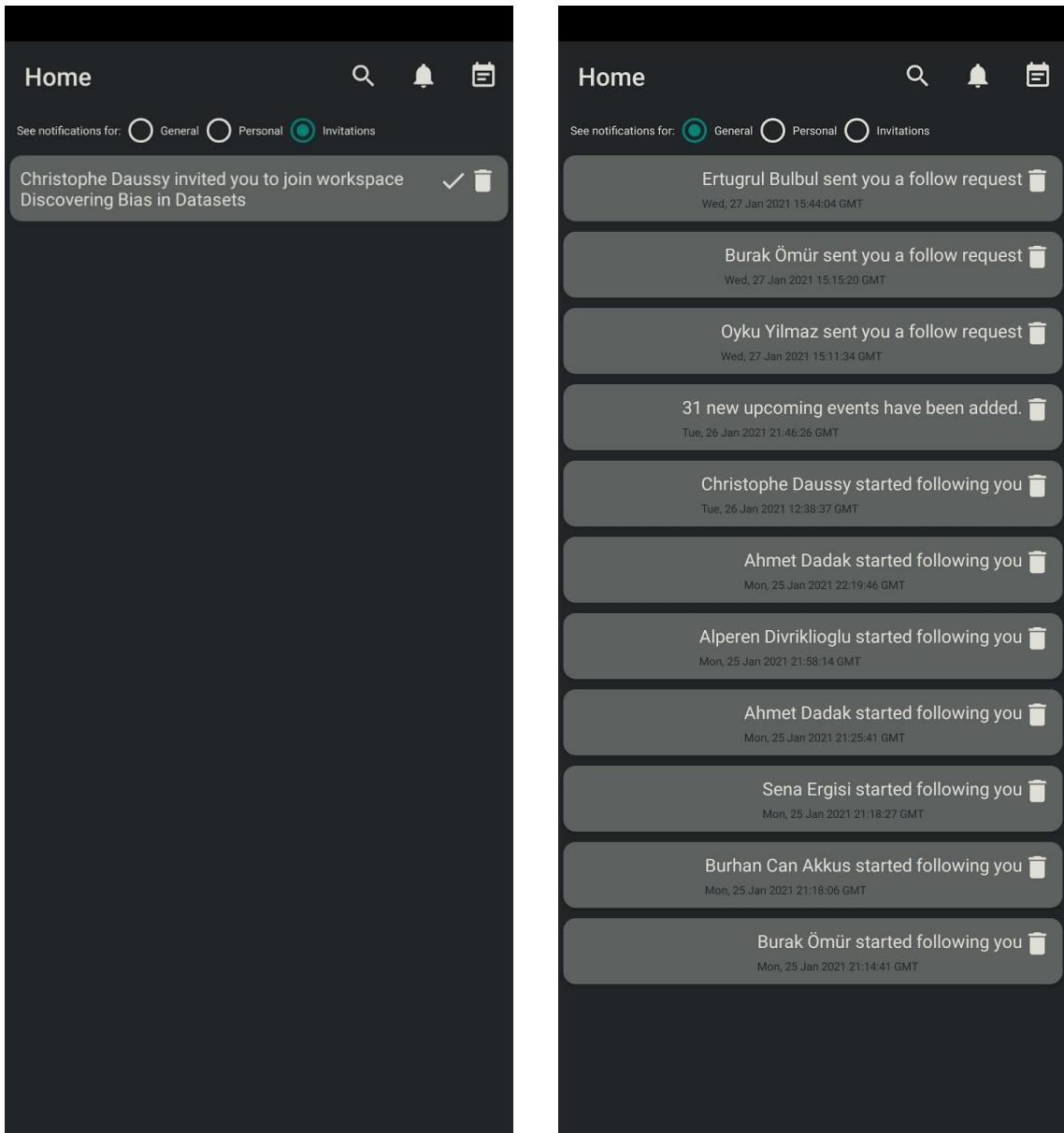


This is how other workspaces look like. You can only view public workspaces. If a workspace interests you, you may send an application by clicking plus button.



If you are the creator of the workspace, you cannot quit it. You have to delete that workspace. If you delete a workspace, you cannot recover it. All the files, milestones, and issues will be deleted. The workspace will automatically disappear in every collaborator's workspace list.

Notifications:



You will receive notifications; you can delete them. Also you can accept workspace invitations in the invitations tab.

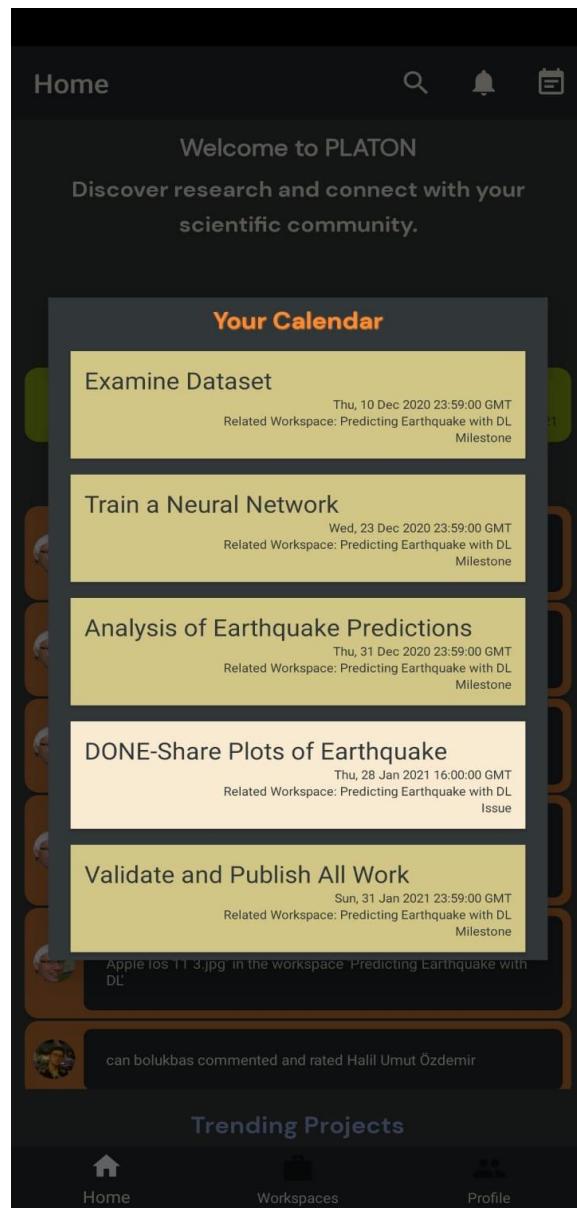
Home Page:



Home page consists of mainly four parts: Upcoming events, Trending Projects, Personal Reminder, and the Activity Stream.

Under this section, trending public workspaces are listed. You may see its details by clicking on the name of the project.

Upcoming Conferences, Journal Special Issues, Submission Deadlines and CFP (call for papers) documents. These are provided by CFP API. If you search for an upcoming event, you may visit its website by clicking on the search result.



Personal reminder consists of the deadlines of issues, milestones, and workspaces which you are a collaborator of. They are listed in ascending order.

Search:

The image displays three separate search results panels, each with a header, search filters, and a list of items.

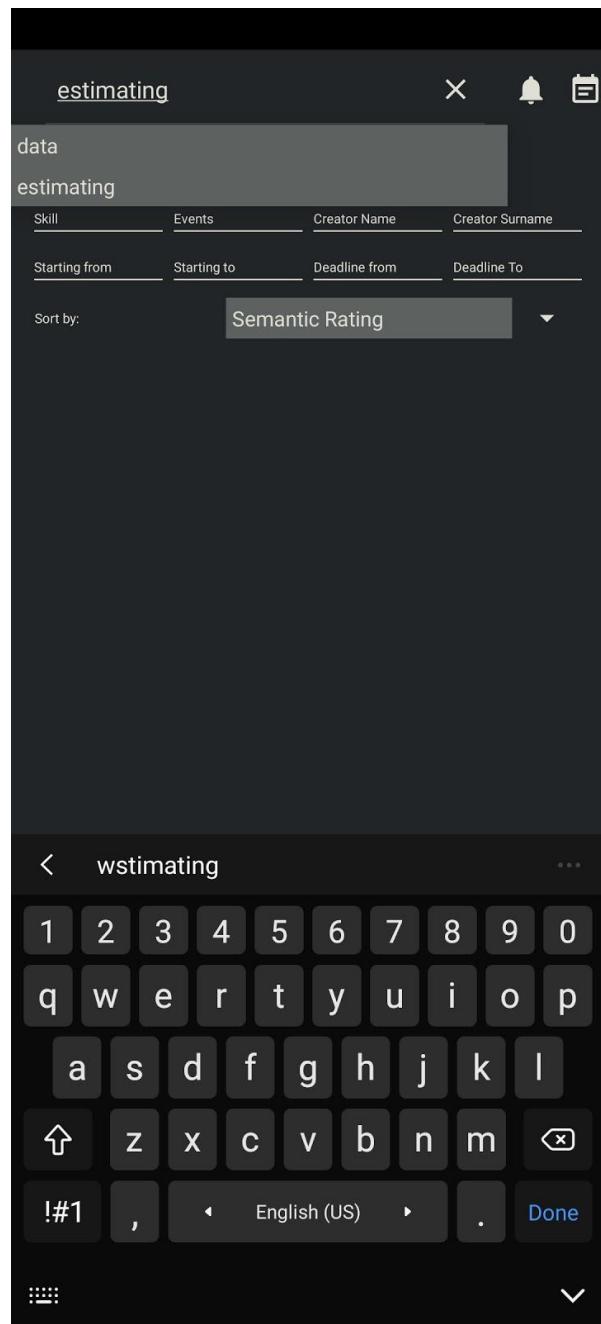
- User Search:** Shows results for "bur".
 - Filter by job: Any
 - Sort by: Semantic Rating
 - Results:
 - Burak Ömür (Privacy: True)
 - Burhan Can Akkus (Privacy: False)
 - Alperen Divriklioglu (Privacy: True)
 - Halil Umut Özdemir (Privacy: False)
- Workspace Search:** Shows results for "ai".
 - Filter by job: Any
 - Sort by: Semantic Rating
 - Results:
 - aiDM 2021 (Jun 5, 2021 - Jun 5, 2021 / Mar 22, 2021 / virtual)
 - TIAD 2021 (Sep 1, 2021 - Sep 1, 2021 / Apr 23, 2021 / Zaragoza, Spain)
 - MIWAI 2021 (Jul 2, 2021 - Jul 3, 2021 / Mar 19, 2021 / Korea (Virtual Conference))
 - MIWAI 2021 (Jul 2, 2021 - Jul 3, 2021 / Mar 19, 2021 / Korea (Virtual Conference))
 - CMCA 2021 (Feb 20, 2021 - Feb 21, 2021 / Jan 31, 2021 / Dubai, UAE)
 - MDAI 2021 (Sep 27, 2021 - Sep 30, 2021 / Mar 22, 2021 / Umea, Sweden)
 - ETAI 2021 (Aug 1, 2021 - Aug 5, 2021 / Feb 3, 2021 / San Diego, California, USA)
 - QUORS 2021 (Jul 13, 2021 - Jul 16, 2021 / Apr 21, 2021 / Virtual due to COVID, Madrid, Spain)
 - JSAI 2021 (Jun 8, 2021 - Jun 11, 2021 / Mar 2, 2021 (Feb 10, 2021) / Online)
 - EALChinacom 2021
- Event Search:** Shows results for "estimating".
 - Filter by job: Any
 - Sort by: Semantic Rating
 - Results:
 - Trejectory Prediction for Autonomous Cars (Mon, 25 Jan 2021 21:21:43 GMT / Thu, 22 Jun 2023 00:00:00 GMT)
 - Predicting Earthquake with DL (Mon, 25 Jan 2021 21:35:15 GMT / Sat, 30 Jan 2021 00:00:00 GMT)
 - Automated Conduction for Symphony Orchestras (Mon, 25 Jan 2021 21:20:49 GMT / Wed, 12 Oct 2022 00:00:00 GMT)
 - Discovering Bias in Datasets (Mon, 25 Jan 2021 21:34:05 GMT / Sun, 28 Mar 2021 00:00:00 GMT)
 - TREC COVID (Mon, 25 Jan 2021 21:34:34 GMT / null)

Platon supports semantic search. Semantic search is a data searching technique in which a search query aims to not only find keywords but to determine the intent and contextual meaning of the words a person is using for search.

You may search for users, workspaces, and upcoming events.

You may use the filters related to every search type to narrow down your search.

Only public workspaces will be listed in the workspace search.



You can see your search history.

SYSTEM MANUAL

BACKEND

Set Up Without Docker-Compose

This project is implemented by using Python 3.8.0 and Flask 1.1.2. To achieve more information about the versions of the libraries that are used, it can be useful to examine "requirements.txt" file in the backend folder of the repository. To comply with compatibility issues, please use the following command to install related libraries.

1. virtualenv <Virtual Environment Name>
2. source <Virtual Environment Name>/bin/activate
3. cd <Path of Repository>/Platon/backend
4. pip/pip3 install -r requirements.txt

To make the database migrations:

5. Assign FLASK_APP environment variable as platon_backend.py
6. Go to <Path of Repository>/platon/backend/config.py file
7. Select the version that you will use.

DEBUG = True and DEVELOPMENT = True => Connects backend to the the MySQL Server in the Live Production Server

DEBUG = False and DEVELOPMENT = True => Connects backend to the the Test MySQL Server

DEBUG = False and DEVELOPMENT = False => This is the Docker-Compose configuration. (Use it only with Docker-Compose Setup)

8. flask db migrate
9. flask db upgrade

To run the backend server:

10. python/python3 platon_backend.py

Set Up with Docker-Compose

It was much easier to set up our backend server by using docker compose. In addition, there is not any external MySQL server required for this setting up process. This commands only runs backend and database servers. To set up with Docker-Compose use the following commands:

1. Install Docker and Docker-Compose
2. cd <Path of Repository>/platon

3. sudo docker-compose up --build (-d flag can be used to run container silently)

To stop all of the containers:

1. cd <Path of Repository>/platon
2. sudo docker-compose down

Set Up All System (Frontend-Backend_Database) With Docker-Compose

In this approach it is possible to set up backend, frontend and database server together. To set up with Docker-Compose use the following commands:

1. Install Docker and Docker-Compose
2. Adjust the Frontend Host Name:

IMPORTANT: This step is only required if you want to deploy frontend into a remote machine. Currently the delivered code is arranged to run in localhost.

Change the FRONTEND_HOSTNAME variable in the <Path of Repository>/platon/config.py file.

Example: FRONTEND_HOSTNAME = "http://ec2-3-120-98-39.eu-central-1.compute.amazonaws.com/"

3. Restore Database Dump(if you want an empty database, skip this step)
 - 3.1 cd <Path of Repository>/platon
 - 3.2 sudo bash restore_database.sh
4. Run All System
 - 4.1 cd <Path of Repository>/platon
 - 4.2 sudo bash run.sh

Database username and password can be found in the path: <Path of Repository>/platon/docker-compose.yml

MYSQL_DATABASE=platondb

MYSQL_ROOT_PASSWORD=rootpassword

MYSQL_USER=root

FRONTEND

Requirement:

You need to install Node.js before you run the following scripts. Your Node.js version must be above 5.0.0.

Recommendation:

In order to have full experience of the app, it is recommended that you should use Google Chrome as the browser.

In the project directory ("platon/frontend" directory), first run:

```
> npm install
```

This installs all the packages in package.json.

To open the app, you need to run:

```
> npm start
```

This runs the app in the development mode. You can open <http://localhost:3000> to view the app in the browser. The page will reload if you make any changes in the code.

To run tests, you need to run:

```
> npm test
```

This runs the provided package's "test" script.

To have production build, you need to run:

```
> npm run build
```

This builds the app for production to the build folder.

Congrats. Your app is ready to be deployed!

This manual is created according to the information on the official create-react-app website. For more detailed information, visit the [website](#). (<https://create-react-app.dev/docs/getting-started/>)

ANDROID

Requirements

- Android Studio
- Android SDK Tools (minSdkVersion:25 targetSdkVersion:29)
- JDK
- Android 6.0 and above (API Level 23)

Setup

Clone the project from Github Repository.

```
- git clone https://github.com/bounswe/bounswe2020group7.git
```

Open the source code in Android Studio.

Install a virtual device on Android Studio or connect an Android phone.

Run the project

Select a virtual device or connected Android phone in Android Studio.

Build and run the project on the selected device.

PROJECT PLAN

		Ad	Süre	Başlat	Bitir...	Önc...	Kaynak Adlar
1		Orientation	6 günler	10.02.2020 17:00	18...		
2		Researching about version control system	6 günler	10.02.2020 17:00	18...		Burak Ömür;Hasan Ramazan Yurt;Halil Umut Özdemir;Öykü Yılmaz
3		Researching about GitHub repositories	6 günler	10.02.2020 17:00	18...		Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil
4		Researching about markdown	6 günler	10.02.2020 17:00	18...		Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil
5		First team meeting	1 gün	13.02.2020 17:00	14...		Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil
6		Creating a workspace on Slack	3 günler	13.02.2020 17:00	18...		Erturul Bülbül
7		Documentation	5 günler	13.02.2020 08:00	19...		
8		Creating the wiki page of the team	3 günler	13.02.2020 08:00	17...		Burak Ömür
9		Creating the readme.md for the repository	4 günler	13.02.2020 08:00	18...		Hasan Ramazan Yurt
10		Preparing communication plan	4 günler	13.02.2020 08:00	18...		Ahmet Dadak
11		Preparing favourite github repositories page	5 günler	13.02.2020 08:00	19...		Meltem Arslan
12		Preparing own personal wiki pages	5 günler	13.02.2020 08:00	19...		Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil
13		Preparing version control system research page	5 günler	13.02.2020 08:00	19...		Öykü Yılmaz
14		Requirements	13 günler	20.02.2020 08:00	09...		
15		Requirement engineering	5 günler	20.02.2020 08:00	26...		Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil
16		Reviewing the requirements before first version	1 gün	28.02.2020 08:00	28...		Halil Umut Özdemir;Meltem Arslan;Burak Ömür;Öykü Yılmaz
17		Adding/altering system requirements	3 günler	28.02.2020 08:00	03...		Meltem Arslan;Burak Ömür;Ahmet Dadak;Kerem Uslular;Hasan Ra...
18		Adding/altering user requirements and non-functional requir...	3 günler	28.02.2020 08:00	03...		Erturul Bülbül;Öykü Yılmaz;Alperen Divrikliolu;Mehmet Temizel;H...
19		Customer meeting #1 for specifying requirements	3 günler	05.03.2020 08:00	09...		Halil Umut Özdemir;Kerem Uslular;Ahmet Dadak
20		Logo and Name	4 günler	28.02.2020 08:00	04...		
21		Preparing logo and name	4 günler	28.02.2020 08:00	04...		Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil
22		User Scenarios	11 günler	28.02.2020 08:00	13...		
23		Creating scenario #1	4 günler	10.03.2020 08:00	13... 14		Öykü Yılmaz;Hasan Ramazan Yurt;Halil Umut Özdemir
24		Creating scenario #2	4 günler	28.02.2020 08:00	04...		Burak Ömür;Alperen Divrikliolu;Erturul Bülbül;Kerem Uslular
25		Creating scenario #3	4 günler	28.02.2020 08:00	04...		
26		Mockups	4 günler	16.03.2020 08:00	19... 22		
27		Creating mockup #1	4 günler	16.03.2020 08:00	19...		Halil Umut Özdemir;Hasan Ramazan Yurt;Öykü Yılmaz
28		Creating mockup #2	4 günler	16.03.2020 08:00	19...		Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Kerem Uslular
29		Creating mockup #3	4 günler	16.03.2020 08:00	19...		
30		Diagrams	11 günler	12.03.2020 08:00	26... 14		
31		Preparing Use Case Diagram	4 günler	12.03.2020 08:00	17...		Ahmet Dadak;Burak Ömür;Meltem Arslan;Erturul Bülbül;Alperen ...
32		Preparing Class Diagram	4 günler	12.03.2020 08:00	17...		Halil Umut Özdemir;Öykü Yılmaz;Kerem Uslular;Mehmet Temizel;H...
33		Preparing Sequence Diagrams	3 günler	24.03.2020 08:00	26... 31...		Burak Ömür;Hasan Ramazan Yurt;Halil Umut Özdemir;Öykü Yılmaz
34		Planning	5 günler	13.04.2020 13:00	20...		
35		Until Planning	4 günler	13.04.2020 13:00	17...		Ahmet Dadak;Alperen Divrikliolu

Platon- Sayfa1

		Ad	Süre	Balat	Bitir...Önc...	Kaynak Adlar
36		After Planning	4 günler	13.04.2020 13:00	17...	Halil Umut Özdemir;Öykü Yılmaz;Meltem Arslan;Kerem Uslular
37		Merge of Plans	1 gün?	17.04.2020 13:00	20... 35...	Burak Ömür;Hasan Ramazan Yurt;Mehmet Temizel;Erturul Bülbül
38		Milestone1	0 günler	04.05.2020 17:00	04... 26...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
39		API Test & Study	12 günler?	23.04.2020 08:00	08...	
40		API Meeting	1 gün?	23.04.2020 08:00	23...	
41		API Usage	5 günler?	23.04.2020 08:00	29...	
42		Design of New API	8 günler?	29.04.2020 08:00	08...	
43		Milestone2	0 günler	18.05.2020 17:00	18... 26...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
44		Arrange teams	1 gün?	27.10.2020 08:00	27...	Burak Ömür;Hasan Ramazan Yurt;Halil Umut Özdemir;Öykü Yılmaz...
45		Revise Requirements, Design, Plans	6 günler	27.10.2020 08:00	03...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
46		Group Meetings	1,833 günler?	04.11.2020 08:00	05...	
47		All members	0,833 günler	04.11.2020 08:00	04...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
48		Backend	1 gün	04.11.2020 15:39	05... 47	Alperen Divrikliolu;Halil Umut Özdemir;Hilal Demir;Hüseyin Can B...
49		Frontend	0,75 günler	04.11.2020 15:39	05... 47	Ahmet Dadak;Burhan Can Akku;Hasan Ramazan Yurt;Umutcan ...
50		Android	0,75 günler	04.11.2020 15:39	05... 47	Burak Ömür;Erturul Bülbül;Orkan Akısu;Öykü Yılmaz
51		Pre Implementation	4,25 günler?	05.11.2020 13:39	11...	
52		Backend	4 günler?	05.11.2020 15:39	11...	
53		Create Test Server	4 günler	05.11.2020 15:39	11... 48	Halil Umut Özdemir;Hüseyin Can Böülüka
54		Frontend	1,5 günler?	05.11.2020 13:39	09...	
55		Create Initial Design	1,5 günler	05.11.2020 13:39	09... 49	Ahmet Dadak;Hasan Ramazan Yurt;Burhan Can Akku;Umutcan ...
56		Android	0,25 günler?	05.11.2020 13:39	05...	
57		Create Initial Design	0,25 günler	05.11.2020 13:39	05... 50	Erturul Bülbül;Burak Ömür;Öykü Yılmaz;Orkan Akısu
58		Test Case	5 günler?	11.11.2020 15:39	18...	
59		Implement Test Cases of Log-in/Register and Profile Page	5 günler	11.11.2020 15:39	18... 53...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
60		Log In - Register	20 günler?	10.11.2020 08:00	07...	
61		Backend	20 günler?	10.11.2020 08:00	07...	
62		Implementation of Register System	4 günler	10.11.2020 08:00	13...	Alperen Divrikliolu
63		Implementation of Log In System	8 günler	16.11.2020 08:00	25... 62	Halil Umut Özdemir
64		Reset-Change Password	8 günler	26.11.2020 08:00	07... 62...	Halil Umut Özdemir
65		Frontend	12 günler?	10.11.2020 08:00	25...	
66		Register Page	12 günler	10.11.2020 08:00	25...	Ahmet Dadak
67		Log In Page	12 günler	10.11.2020 08:00	25...	Ahmet Dadak
68		Reset-Change Password Page	6 günler	10.11.2020 08:00	17...	Ahmet Dadak
69		Android	12 günler?	10.11.2020 08:00	25...	
70		Register Page	12 günler	10.11.2020 08:00	25...	Burak Ömür

Platon- Sayfa2

		Ad	Süre	Balat	Bitir...Önc...	Kaynak Adler
71		Log In Page	12 günler	10.11.2020 08:00	25...	Burak Ömür
72		Reset- Change Password Page	6 günler	10.11.2020 08:00	17...	Burak Ömür
73		Profile Page - User Actions	20 günler	16.11.2020 08:00	11...	
74		Backend	20 günler	16.11.2020 08:00	11...	
75		Account Information System	4 günler	08.12.2020 08:00	11... 62...	Alperen Divrikliolu;Halil Umut Özdemir;Hilal Demir
76		User Action - Following System	8 günler	16.11.2020 08:00	25...	Hüseyin Can Böyükba
77		User Action - Reporting System	3 günler	16.11.2020 08:00	18...	Alperen Divrikliolu;Halil Umut Özdemir;Hilal Demir;Hüseyin Can B...
78		Profile Privacy	4 günler	16.11.2020 08:00	19...	Halil Umut Özdemir
79		Linking Google Scholar-ResearchGate Account	12 günler	16.11.2020 08:00	01...	Halil Umut Özdemir
80		Frontend	12 günler	16.11.2020 08:00	01...	
81		User Profile Page	12 günler	16.11.2020 08:00	01...	Hasan Ramazan Yurt
82		Follower and Following Accounts Pages	6 günler	16.11.2020 08:00	23...	Hasan Ramazan Yurt
83		Follow Requests Pages	6 günler	16.11.2020 08:00	23...	Hasan Ramazan Yurt
84		Reporting Page	3 günler	16.11.2020 08:00	18...	Hasan Ramazan Yurt
85		Android	4 günler	16.11.2020 08:00	19...	
86		User Profile Page	4 günler	16.11.2020 08:00	19...	Burak Ömür;Erturul Bülbül;Öykü Yılmaz
87		Follower and Following Accounts Pages	3 günler	16.11.2020 08:00	18...	Burak Ömür;Erturul Bülbül;Öykü Yılmaz
88		Follow Requests Pages	1 gün	16.11.2020 08:00	16...	Burak Ömür;Erturul Bülbül;Öykü Yılmaz
89		Reporting Page	1 gün	16.11.2020 08:00	16...	Burak Ömür;Erturul Bülbül;Öykü Yılmaz
90		Testing	1 gün	16.11.2020 08:00	16...	
91		Test Log in-Register and Profile Page	1 gün	16.11.2020 08:00	16... 62	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
92		Internal Milestone 1	0 günler	11.12.2020 17:00	11... 58...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
93		Milestone3	0 günler	11.12.2020 17:00	11... 51...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
94		Search Engine	9 günler	30.11.2020 08:00	10...	
95		Backend	8 günler	30.11.2020 08:00	09...	
96		Basic User- Workspace- Event Search (Semantic)	2,667 günler	30.11.2020 08:00	02...	Halil Umut Özdemir;Hilal Demir;Hüseyin Can Böyükba
97		Advanced User- Workspace- Event Search (Semantic)	4 günler	30.11.2020 08:00	03...	Halil Umut Özdemir;Hilal Demir;Hüseyin Can Böyükba
98		Search History	4 günler	04.12.2020 08:00	09... 96...	Halil Umut Özdemir
99		Frontend	9 günler	30.11.2020 08:00	10...	
100		Basic User- Workspace- Event Search Section	6 günler	30.11.2020 08:00	07...	Hasan Ramazan Yurt
101		Advanced User- Workspace- Event Search Section	9 günler	30.11.2020 08:00	10...	Hasan Ramazan Yurt
102		Search History Section	6 günler	01.12.2020 08:00	08...	Hasan Ramazan Yurt
103		Search Results Page	3 günler	03.12.2020 08:00	07...	Hasan Ramazan Yurt
104		Android	9 günler	30.11.2020 08:00	10...	
105		Basic User- Workspace- Event Search Section	6 günler	30.11.2020 08:00	07...	Burak Ömür

Platon- Sayfa3

		Ad	Süre	Balat	Bitir...Önc...	Kaynak Adalar
106		Advanced User- Workspace- Event Search Section	9 günler	30.11.2020 08:00	10...	Burak Ömür
107		Search History Section	3 günler	02.12.2020 08:00	04...	Burak Ömür
108		Search Results Page	6 günler	03.12.2020 08:00	10...	Burak Ömür
109		Testing	0,833 günler	07.12.2020 08:00	07...	
110		Implement Test Cases of Search Engine	0,833 günler	07.12.2020 08:00	07...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Burhan Can Akku;...
111		Test Case	4 günler	07.12.2020 08:00	10...	
112		Implement Test Cases of Search Engine	4 günler	07.12.2020 08:00	10...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
113		Workspace	21 günler	03.12.2020 08:00	31...	
114		Backend	20 günler	04.12.2020 08:00	31...	
115		Workspace Information System	12 günler	04.12.2020 08:00	21...	Alperen Divrikliolu
116		File Storage System	8 günler	22.12.2020 08:00	31... 115	Halil Umut Özdemir
117		Issue - Milestone System	8 günler	22.12.2020 08:00	31... 115	Hüseyin Can Bölükba
118		Comment-Rate System	8 günler	22.12.2020 08:00	31... 115	Hilal Demir
119		Invitation Mechanism	4 günler	22.12.2020 08:00	25... 115	Alperen Divrikliolu
120		Frontend	12 günler	03.12.2020 08:00	18...	
121		Workspace Pages for all States of Workspaces	12 günler	03.12.2020 08:00	18...	Ahmet Dadak
122		Edit File Page	6 günler	03.12.2020 08:00	10...	Ahmet Dadak
123		Issue - Milestone Pages	4 günler	03.12.2020 08:00	08...	Burhan Can Akku;Umutcan Uvut
124		Add Comment Rate Sections to Profile Pages	0 günler	03.12.2020 08:00	03...	Hasan Ramazan Yurt
125		Invitation Section of Workspace	2 günler	03.12.2020 08:00	04...	Ahmet Dadak
126		Add Invitation Section to User Profiles	1 gün	03.12.2020 08:00	03...	Ahmet Dadak;Umutcan Uvut
127		Android	12 günler	03.12.2020 08:00	18...	
128		Workspace Pages for all States of Workspaces	12 günler	03.12.2020 08:00	18...	Öykü Yılmaz
129		Edit File Page	3 günler	03.12.2020 08:00	07...	Burak Ömür;Öykü Yılmaz
130		Issue - Milestone Pages	1,5 günler	03.12.2020 08:00	04...	Erturul Bülbül;Öykü Yılmaz
131		Add Comment Rate Sections to Profile Pages	3 günler	03.12.2020 08:00	07...	Burak Ömür
132		Invitation Section of Workspace	1,5 günler	03.12.2020 08:00	04...	Burak Ömür;Öykü Yılmaz
133		Add Invitation Section to User Profiles	1,5 günler	03.12.2020 08:00	04...	Burak Ömür;Öykü Yılmaz
134		Testing	2 günler	15.12.2020 08:00	16...	
135		Test Upcoming Events-Workspace	2 günler	15.12.2020 08:00	16...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
136		Upcoming Events	16 günler	21.12.2020 08:00	11...	
137		Backend	16 günler	21.12.2020 08:00	11...	
138		Upcoming Events API	16 günler	21.12.2020 08:00	11...	Halil Umut Özdemir
139		Frontend	6 günler	21.12.2020 08:00	28...	
140		Upcoming Event Page	6 günler	21.12.2020 08:00	28...	Umutcan Uvut

Platon- Sayfa4

		Ad	Süre	Balat	Bitir... Önc...	Kaynak Adalar
141		Upcoming Events Calendar Widget	6 günler	21.12.2020 08:00	28...	Umutcan Uvut
142		Android	3 günler	21.12.2020 08:00	23...	
143		Upcoming Event Page	3 günler	21.12.2020 08:00	23...	Burak Ömür;Öykü Yılmaz
144		Upcoming Events Calendar Widget	3 günler	21.12.2020 08:00	23...	Burak Ömür;Öykü Yılmaz
145		Internal Milestone 2	0 günler	22.12.2020 08:00	22...	
146		Test Case	3 günler	24.11.2020 08:00	26...	
147		Implement Test Cases of Upcoming Events-Workspace	3 günler	24.11.2020 08:00	26...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
148		Milestone4	0 günler	11.01.2021 17:00	11... 11...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
149		Test Case	3,333 günler	31.12.2020 08:00	05...	
150		Implement Test Cases of Recommendation System-Home Pag	3,333 günler	31.12.2020 08:00	05...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Burhan Can Akku;...
151		Recommendation System	3 günler	05.01.2021 08:00	07...	
152		Backend	3 günler	05.01.2021 08:00	07...	
153		Recommendation System	3 günler	05.01.2021 08:00	07...	Halil Umut Özdemir
154		Frontend	3 günler	05.01.2021 08:00	07...	
155		Add Recommendation Section to Profile Pages	2 günler	05.01.2021 08:00	06...	Hasan Ramazan Yurt
156		Add Recommendation Section to Workspace Pages	3 günler	05.01.2021 08:00	07...	Ahmet Dadak
157		Android	3 günler	05.01.2021 08:00	07...	
158		Add Recommendation Section to Profile Pages	3 günler	05.01.2021 08:00	07...	Burak Ömür;Erturul Bülbül;Öykü Yılmaz
159		Add Recommendation Section to Workspace Pages	2 günler	05.01.2021 08:00	06...	Burak Ömür;Erturul Bülbül;Öykü Yılmaz
160		Home Page	9 günler	07.01.2021 08:00	19...	
161		Backend	3 günler	07.01.2021 08:00	11...	
162		Email Notification System	2 günler	07.01.2021 08:00	08...	Alperen Divrikliolu
163		Activity Stream System	3 günler	07.01.2021 08:00	11...	Hüseyin Can Böyükba
164		Frontend	9 günler	07.01.2021 08:00	19...	
165		Home Page	9 günler	07.01.2021 08:00	19...	Burhan Can Akku
166		Add Upcoming Event Calendar Widget	3 günler	07.01.2021 08:00	11...	Umutcan Uvut
167		Android	9 günler	07.01.2021 08:00	19...	
168		Home Page	9 günler	07.01.2021 08:00	19...	Erturul Bülbül
169		Add Upcoming Event Calendar Widget	1 gün	07.01.2021 08:00	07...	Burak Ömür;Erturul Bülbül;Öykü Yılmaz
170		Testing	1 gün	07.01.2021 08:00	07...	
171		Test Recommendation System and Home Page	1 gün	07.01.2021 08:00	07...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Erturul Bülbül;Halil...
172		System Testing	6 günler?	10.01.2021 08:00	18...	
173		Internal Milestone 3	0 günler	19.01.2021 17:00	19... 94...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Burhan Can Akku;...
174		Milestone5 (Final Milestone)	0 günler	19.01.2021 17:00	19... 15...	Ahmet Dadak;Alperen Divrikliolu;Burak Ömür;Burhan Can Akku;...

Platon- Sayfa5

LESSONS LEARNED FROM CUSTOMER MEETING

Since we have prepared a well-thought scenario for our final presentation, it went very smooth. We have managed to display all the features, pages, functionalities and mechanisms we have implemented since Milestone 4. We did not encounter any bugs or problems while we were presenting our web and mobile applications. The general feedback for our presentation was also good.

We have displayed all the features and we have applied the project plan accordingly so far. Generally, our customer seemed satisfied by our current status since there was not a major problem or failure of fulfillment was issued.

From customer feedback, we saw that our file storage mechanism is a little problematic since when a user deletes a file, the file is gone and there is no way to revert that back. In the applications, there is a warning system for that. In the future, we can improve the mechanism with a git-based repository system. It would be better to give the chance of reverting back for these kinds of situations.