
CEng 491 - Fall 2020

First lecture

Course staff

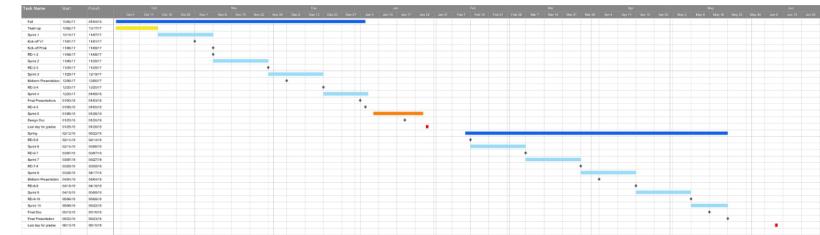
Pelin Angın, Gökberk Cinbiş

(Coordinators)

Abdullah Doğan, Alper Karamanlıoğlu, Barış Nasır, Anıl Çetinkaya, Kadir Cenk Alpay, Nermin Samet, Çağlar Seylan, Doğuhan Yeke

(Teaching Assistants)

Senior Design Project



- 491+492 as a whole: a (almost) real-life project development experience
- Application of your basic software & hardware engineering knowledge to a real-life problem/project.
- Team work
 - 4 developers
 - 1 supervisor (acting as the development manager/boss)
 - 1 communication assistant
 - ++ assistants/ coordinators
- Spans 2 semesters, 4+4 credits !!!!
 - Expectations are higher!
 - Both the quality and the quantity of your work are important!

Course characteristics

Agile Methodology

Iterations (Sprints)

- Plan what to do
- Do what you plan
- Revise your plan (+/- todo items)
- Iterate

There will be a lecture about agile development.

Course characteristics (cont'd)

Each week we have 2 lecture hours (Thursday 13:40-15:30) + **4 lab hours (Wednesday 13:40-17:30)**.

Lecture hours are for:

- Lectures
- Seminars
- Team presentations (midterm, coordinator meetings)

Lab hours are reserved for weekly meetings. Two (or one) meetings:

- With your assistant and your supervisor.

Course characteristics (cont'd)

Work will be done in 3-week sprints.

We will have 4 sprints this semester. (Check the syllabus for start/end dates)

There will be in-class presentations. These will be considered as oral exams:

- 1st: 2 min. elevator pitch per group, just after submitting the first kick-off document draft.
- 2nd: towards the end of the semester.

There will be coordinator meetings. These will be considered as oral exams as well.

Course characteristics (cont'd)

Documentation: We try to keep it at a minimum.

- You will deliver a **kick-off document** at the beginning
- and a **design document** at the end of the semester.
- You will deliver a **retrospective document** at the beginning of each sprint.

Timeline

	Start	End	Retrospective Deadline
Sprint 1	28-Oct	17-Nov	18-Nov
Sprint 2	18-Nov	8-Dec	9-Dec
Sprint 3	9-Dec	29-Dec	30-Dec
Sprint 4	30-Dec	19-Jan	20-Jan

Deliverable	Tentative Deadline
Kickoff Doc V1	9-Nov
Elevator pitch	12-Nov
Kickoff Final	18-Nov
Design Doc	18-Jan
Midterm Presentation	6-7 Jan
Coordinator Meetings	To be announced

Course characteristics (cont'd)

Classical course vs 49x

- In a classical undergrad course, there are well-defined assignments and written-exams. Their answers/solutions are known.
- In 49x, answers/solutions are not known or defined!
- This creates a difficulty for grading.
- 491 grading more subjective. We always try to reduce this.
- So, focus on the total quality of your work rather than individual grading items.

Seminars

- Technical/Technological topics (Hardware, Software)
- Project Management
- (Engineering) Ethics, Social Responsibility
- Innovation
- Entrepreneurship
- Intellectual Property (Patents, etc.)
- Open Source
- Professional Society
- And more...

Grading

ITEM	PERCENTAGE
Weekly Engineering Practice	40
Sprint performance	20
Kick-off	10
Midterm exam	15
Design Document	10
Online Seminar Attendance	5
Total	100

Grading

% 40 --- Weekly engineering practice (WEP) - by your TA

Criteria overview (for individuals):

- The general progress
- The amount of work done
 - The quantitative and qualitative usage of GitLab
- The meeting agendas (GitLab)
- Attendance to the meetings
- Individual level gradings

Grading

- There is no group level grading in WEPs, but we will be following your group progresses.
- Your group performance also will be evaluated through midterm, coordinator meetings, etc.

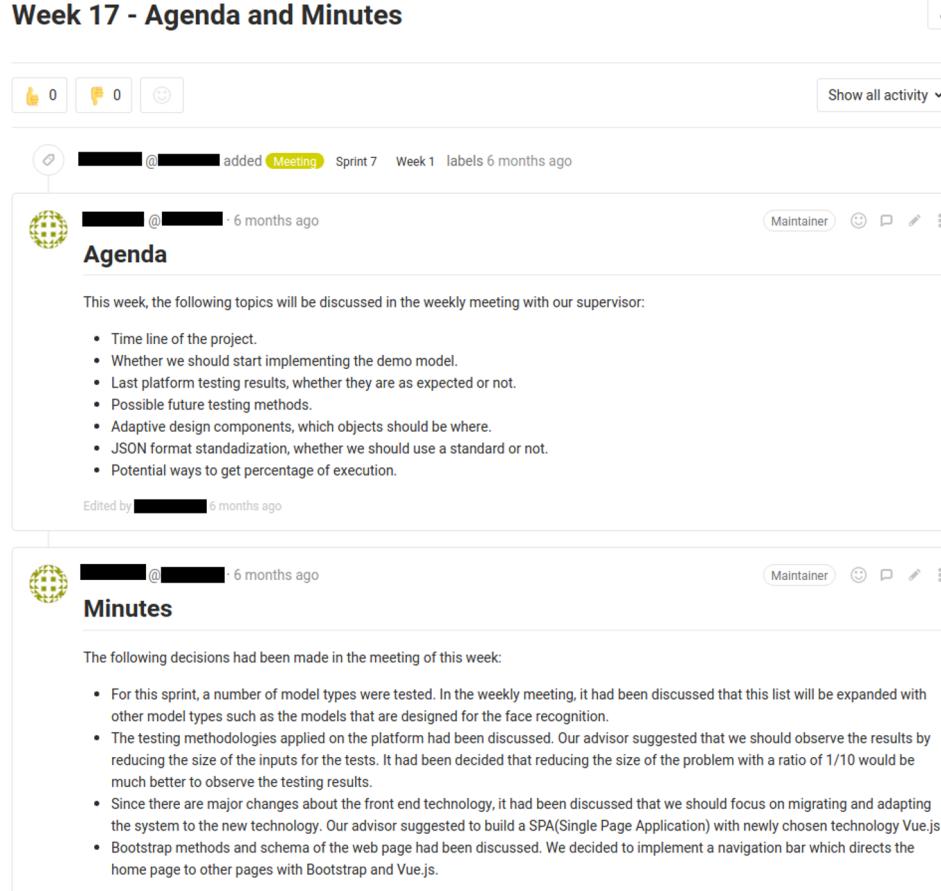
G4	Supervisor Meeting Agenda and Minutes
I1	Attendance (When called)
I2	Coding / Research Activity
I3	Fake Activity
I4	Weekly Gitlab Activity as Issues (Open/Update/Close)
I5	Weekly Workload

Grading Examples Issues

Week X	Oct. 2	Wednesday	Open the issues of Week X. Supervisor meetings of Week X.
	Oct. 3	Thursday	
	Oct. 4	Friday	
	Oct. 5	Saturday	
	Oct. 6	Sunday	
	Oct. 7	Monday	
	Oct. 8	Tuesday	Close the issues of Week X until 23:55
	Oct. 9	Wednesday	We will grade Week X
●	Oct. 10	Thursday	We will grade Week X
	Oct. 11	Friday	We will grade Week X
	Oct. 12	Saturday	
	Oct. 13	Sunday	
	Oct. 14	Monday	Week X grades will be announced. You should send an e-mail to 49x about your objections.
	Oct. 15	Tuesday	You should send an e-mail to 49x about your objections.
	Oct. 16	Wednesday	Objections and meetings of Week X, No further objections will be taken into account after this day. You must attend the meetings.

Grading Examples with Snapshots - Agenda and Minutes

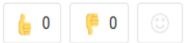
Week 17 - Agenda and Minutes



Grading Examples with Snapshots - Agenda and Minutes

CANCELLED MEETING - Week 18 - Agenda and Minutes

Edited 6 months ago by [REDACTED]



Show all activity ▾

[REDACTED] @ [REDACTED] added Meeting Sprint 7 Week 1 labels 6 months ago



[REDACTED] @ [REDACTED] · 6 months ago

Maintainer



Agenda

This week, the following topics will be discussed in the weekly meeting with our supervisor:

- Latest testing results.
- Current logging implementation and alternative methods.
- Time line of the project.
- Whether we should start implementing the demo model.

Edited by [REDACTED] 6 months ago



[REDACTED] @ [REDACTED] · 6 months ago

Maintainer



Minutes

Meeting was cancelled due to scheduling issues. These topics will be discussed in following meeting.

Edited by [REDACTED] 5 months ago

- It is your responsibility to setup weekly meetings with your supervisor.
- When a meeting with your supervisor is cancelled, you need to add a note on that.
- Remember that any misleading information will be punished.

Grading Examples with Snapshots - Issue Tags

Investigation about Frontend Technologies React and Vue.js #64 · opened 6 months ago by [REDACTED]	Done Interface	Sprint 6 Week 2	CLOSED	  1  5	updated 6 months ago
Week 16 - Agenda and Minutes #65 · opened 6 months ago by [REDACTED]	Meeting	Sprint 6 Week 3	CLOSED	  1  3	updated 6 months ago
Week 15 - Agenda and Minutes #60 · opened 6 months ago by [REDACTED]	Meeting	Sprint 6 Week 1	CLOSED	  3	updated 6 months ago
Evaluation of ONNX as Transportation Format and Execution Backend Testing #59 · opened 6 months ago by [REDACTED]	Done Execution	Sprint 6 Week 1	CLOSED	  2	updated 6 months ago
Investigation on ONNX Models and PyTorch #61 · opened 6 months ago by [REDACTED]	Done Execution	Sprint 6 Week 1	CLOSED	 5	updated 6 months ago
Implementing CRUD Operations of the Front End #58 · opened 6 months ago by [REDACTED]	Interface	Sprint 6 Week 1	CLOSED	  5	updated 6 months ago
Changing the Implementation of the Admin Page #43 · opened 9 months ago by [REDACTED]	Done Interface	Sprint 6 Week 1	CLOSED	  5	updated 6 months ago
Model Retrieval and Execution Scripts #38 · opened 9 months ago by [REDACTED]	Done Execution	Sprint 3 Week 2	CLOSED	 8	updated 6 months ago
Implementation of Terminal Client #54 · opened 7 months ago by [REDACTED]	Done General	Sprint 5 Week 3	CLOSED	  2	updated 6 months ago
Week 14 - Agenda and Minutes #57 · opened 7 months ago by [REDACTED]	Meeting	Sprint 6 Week 1	CLOSED	 3	updated 6 months ago

Grading Examples with Snapshots - Issue Title, Description, Commit

Implementation of Transportation of Newly Added Models

This week, I will work on implementing a transportation method to transport newly added models to resources. This will make it possible to send new models from frontend to resources.

The screenshot shows a GitHub pull request interface with four commit sections:

- Introduction**:
As per our last meeting with our advisor, we decided that we should be able to add files when adding a new model to the system. This consists of two main tasks, of which one of them I am doing this week. First task is *transporting files for newly added models*, and second is *registering newly added models to database and triggering an update*. This week, I will work on transporting files for newly added models. Main obstacles about this is the file sizes for huge models, and sending these files synchronously to resources.
- Work**:
Initially, I have decided to implement command control logic in interface manager. This is the point where messages from frontend are received by main platform and processed. Here, I had to decide which method to use to decide command types, such as add or get. I have decided to add new commands to message object, so that content can be used for message specific information, such as which model to add or which model to delete. Then, I had to decide how to process these messages. I have decided to use methods in storage manager, so that single responsibility principle will hold. This made both the code look better, and logic to stay consistent. My work can be seen in [this commit](#).
- Summary**:
This week, I have implemented command control logic for interface manager, and fixed some previous bugs that were in various modules in our platform. My future work will include implementing refresh (update) mechanisms for managers after changes in database, and implementing backend logic for sending these messages. My work can be seen in commits below.
PS: I have written a longer report on this subject, but gitlab seems to disagree with me on comment edits. I cannot seem to be able to retrieve my edits after page update (as it happens after sleep mode on my computer etc.).
- Commits**:
 - abdfbf33
 - a5f322ef
 - a744675d

Grading Examples with Snapshots - Issue Title, Description, Commit

Refactoring and Implementing Front End Operations

We decided to make some changes about front end side after this week's meeting with our supervisor. This week I will work on what to do to make the website more attractive.

parent f9a2ad91 ¶ master
No related merge requests found

Changes 6

Showing 6 changed files ▾ with 96 additions and 73 deletions

backend/configuration.yml

```
... @@ -2,7 +2,8 @@ Models:  
 2 # Model  
 3 - name: "Example Model 1"  
 4   path: "/path/to/model1"  
 5 - type: "image"  
 5 +   inp_type: "image"  
 6 +   out_type: "text"  
 7   description: |  
 8     This is model is an example model for testing out the platform.  
 9   Do not try to execute this. This does not work.  
... @@ -12,7 +13,8 @@ Models:  
 12 # Model  
 13 - name: "Example Model 2"  
 14   path: "/path/to/model2"  
 15 - type: "image"  
 16 +   inp_type: "image"  
 17 +   out_type: "text"  
 18   description: |  
 19     This is model is an example model for testing out the platform.  
 20   Do not try to execute this. This does not work.  
... 
```

During my investigation I realized that people generally used Vue with Flask instead of React.
Therefore, me and my team-mates discussed about whether to use Vue or React. Our team-mate Misra will investigate this matter deeply this week. After this investigation, we will make related changes for the chosen technology at the front end side.

In last week's commit, I implemented CRUD operations to edit and delete selected model and resources.
One missing part was URL converters.
I should take selected model or resource identifier as a argument from the URL.

```
@app.route('/page/<int:page_id>')  
def page(page_id):  
    # Related code
```

I also did some refactoring related to the model objects.
Previously, input and output types were kept as one variable.
I distinguished input and output types to handle them more easily.

My commits for this week can be seen below:

- 4e225564
- 9c1d684
- cc419089

closed 6 months ago

49x Communication @49x-info · 6 months ago
read

Refactored model types by separating input and output types.

This commit includes changes related to the model object. With this commit it is more clear to distinguish input and output types separately.

parent f9a2ad91 ¶ master
No related merge requests found

Changes 6

Showing 6 changed files ▾ with 96 additions and 73 deletions

View file @ 4e225564

backend/diem.py

```
... @@ -325,7 +325,8 @@ def start(manager_address):  
 325   objects.Model  
 326   netconf(model_ids),  
 327   model_path),  
 328 - model("type")  
 328 +   model["inp_type"],  
 329 +   model["out_type"]  
 330   )  
 331   for model in config["Models"]  
 332   ... 
```

View file @ 4e225564

backend/managers/storage_manager.py

```
... @@ -78,7 +78,8 @@ class StorageManager(threading.Thread, metaclass=singleton.Singleton):  
 78   model = objects.Model  
 79   result["id"],  
 80   result["path"],  
 81 - result["type"]  
 81 +   result["inp_type"],  
 82 +   result["out_type"]  
 83   )  
 84   return model  
 85   ... 
```

... @@ -89,7 +89,8 @@ class StorageManager(threading.Thread, metaclass=singleton.Singleton):
 89 objects.Model

Please do NOT link online libraries !

Grading Examples with Snapshots - Research Issues

Research on Spatial Database (PostGIS)



PostGIS is a spatial database extender for PostgreSQL (relational database) and it adds spatial functions such as distance, area, union, intersection, and specialty geometry data types to the database.

Our project will use this technology as database to keep geographic data.



Show all activity ▾

- No explanation
- No code
- No comparison

 [REDACTED] @ [REDACTED] · 10 months ago

Maintainer    

pgAdmin installed to control data.

 [REDACTED] @ [REDACTED] closed 10 months ago

 49x Communication @49x-info · 10 months ago

Owner    

- Not enough explanation
- No research
- No coding activity

Grading Examples with Snapshots - Research Issues

Semi-Supervised Learning Research

This week, I intend on looking into semi-supervised learning starting from `sklearn`. Right now we are using unsupervised, the results need more testing however haven't been what we are shooting for. If we can integrate a semi-supervised approach, we might get better results. I will be looking into how we can do that and do that easily.

Edited 5 months ago by [REDACTED]



[REDACTED] @ [REDACTED] changed due date to April 16, 2019 5 months ago

[REDACTED] @ [REDACTED] added `semi` label 5 months ago

[REDACTED] @ [REDACTED] changed the description 5 months ago

[REDACTED] @ [REDACTED] 5 months ago

Finding a Basis

Commit Link

In order to be able to compare semi-supervised approach with unsupervised and supervised methods, I wanted to find a base. I ended up trying out different algorithms for base: AutoEncoder, RandomForest and 4 other unsupervised learning algorithms we already have working on our detection server.

Right now, other group members are vectorizing and labeling the new data. For these experiments I am using the old one, we can easily run experiments with new data when it is prepared.

Using the homogeneous data, I generated random indices to select train and test sets. This is done so that data is 'shuffled' and both train and test sets contain inliers and outliers. For my experiments, I took 80% of the data to be in train set and remaining 20% in the test set.

I prepared myself a small report with the outputs. In my commit, the code generating the report can be found. Here is the summary of the results:

Algorithm	Accuracy
AutoEncoder	0.74
RandomForest	0.99
OneClassSVM	0.74
IsolationForest	0.67
LocalOutlierFactor	0.68
MCD	0.74

As the supervised base, I chose `RandomForest`, and as for the unsupervised case, I chose `OneClassSVM`.

Edited 5 months ago by [REDACTED]

[REDACTED] @ [REDACTED] 5 months ago

Semi-Supervised Tests

Commit Link

`sklearn` has two semi-supervised learning algorithms already implemented, these are `LabelPropagation` and `LabelSpreading`. In order to simulate the seminess, I randomly selected some points and assigned them to -1 (each vector has to have a label, for unlabeled entries -1 is the identifier). I made it so that you can enter different ratios of labeled/unlabeled data.

In the experiments, I have obtained some unexpected results. Even with the ratio being extremely small (very little labeled data), the models for both algorithms achieved 99% accuracy, precision, and recall. I looked for a possible coding error that I made however variables seemed correct. It is very strange and my only explanation is that data is extremely linear so even with a few labels, correct classifier can be obtained by model. I intend on finding other semi-supervised algorithms from different libraries and try them out. If the same problem occurs, we should really not bother with this dataset anymore.

- Good research issue example
- Clear explanation
- Nice comparison table
- Integrating research into project
- Implementation and initial results

Grading

%20 --- Sprint performance - by Coordinators (Pelin & Gökberk)

Criteria overview (for the group and individuals):

- Retrospective document
- Progress of the team (if necessary individual grading)
- The accuracy of the sprint summary and the upcoming sprint plan
- **Peer grades** in group will significantly affect the final sprint performance grade (it will be collected at the end of the semester).
- **Feedback from supervisors** will also be taken into account.

Grading

In class presentations:

- %15 midterm

Documents:

- %10 kickoff document
- %10 design document

ITEM	PERCENTAGE
Weekly Engineering Practice	40
Sprint performance	20
Kick-off	10
Midterm exam	15
Design Document	10
Attendance	5
Total	100

Syllabus

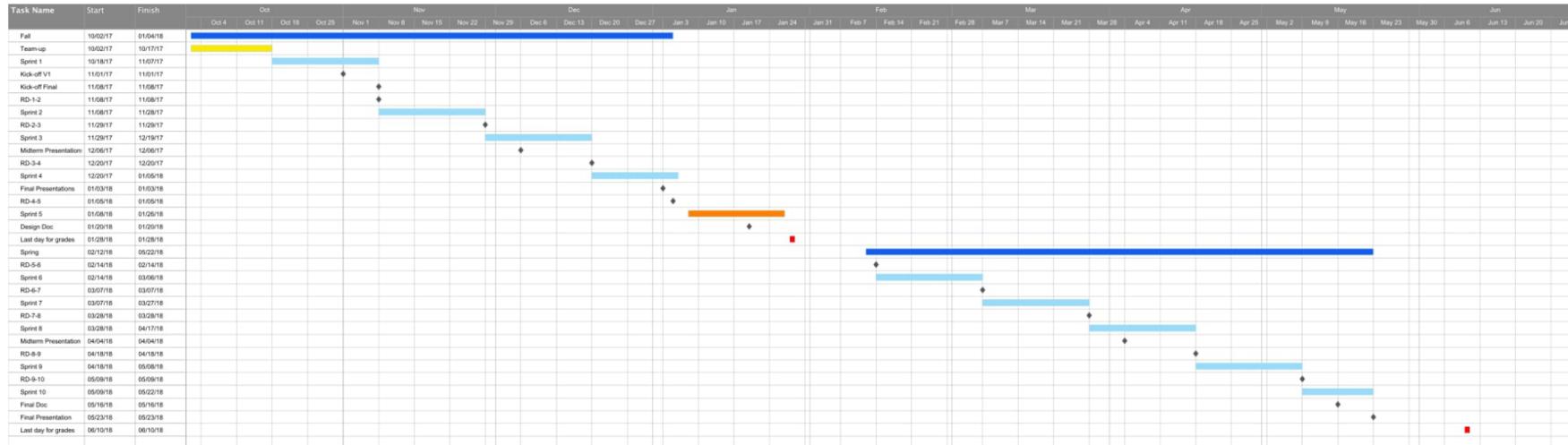
<https://docs.google.com/document/d/e/2PACX-1vTlThXaAd13cpugAxwVD85o8kl2-fygU7RsMJUVuv1yFCfVDRFNhasAtW1JeeSc9ROdUROAB1EeuDU1/pub>

Team-up

Upload your team info to ODTUClass by **October 18**

- List of accepted proposals: [FinalProposals-2020-2021](#)
- First-round deadline for submitting your own proposal: **October 19 @noon**
- First-round deadline for submitting your project preferences (on ODTUClass): **October 20 @noon**
- First round project-team assignments: **October 21**
- Second-round deadline for submitting your proposal or project preference: **October 24@23:59**
- Final project-team assignments: **October 26**
- First sprint starts: **October 28**
- Communication TAs will be announced: **October 28**

Throughout the academic year



At the end, DEMO day in June 2021

Demo day: Great opportunity for you to show your work...

<https://senior.ceng.metu.edu.tr/2020/mainpage/>

- Check the projects from previous years

Evaluation criteria in the next two slides.



Evaluation

Please evaluate the selected project using a linear scale of 1 to 5.

Innovation *

Please evaluate the innovative aspect of the project.

1

2

3

4

5

Very low

Very high

Technical Quality and Effort *

Please evaluate the technical (engineering design and development) quality of the project and the amount of effort that you perceive to be put into developing it.

1

2

3

4

5

Very low

Very high

Potential Impact *

Please evaluate the potential impact of the project to the society, industry or academia.

1

2

3

4

5

Very low

Very high

Demo *

Please evaluate the quality of the demo.

1

2

3

4

5

Very low

Very high

Questions?