

System Specification V.3

Group 13

Background

We need to design a system that simplifies communication within a university course lab. This system should manage time and tasks, track team progress, monitor student attendance, and collect information on task completion time. Additionally, the lab requires a digital solution for administering mandatory tests and real-time data collection. A well-organized and user-friendly system is necessary, but its effectiveness must be verified with feedback from students, teachers, and teaching assistants.

In the lab, students work in teams, and the teachers and TAs need to know how the teams are progressing, which teams are missing members, and how far each group is in their coursework. The system should also collect information on task completion time to help the professor make changes to the workload for the following academic year.

To avoid using large quantities of paper, a digital solution is preferable for managing the lab's many components. By digitizing the system and combining all necessary features, we can simplify the workload for TAs and teachers. However, creating such a system can be challenging, and its effectiveness must be verified through student, teacher, and TA feedback.

Vision

Our vision is to create a user-friendly webpage that centralizes the needs of students, teaching assistants, and the teacher.

The RATs will be digitized to enable easy access to past tests and real-time statistics on results. This will allow the teacher to identify difficult questions and adjust the coursework accordingly.

We will provide a timer for team management, which each team can activate individually to manage their breaks as they prefer.

By digitalizing the RATs, the teacher to monitor completion of the RATs and track team progress in completing the lab tasks.

Objectives

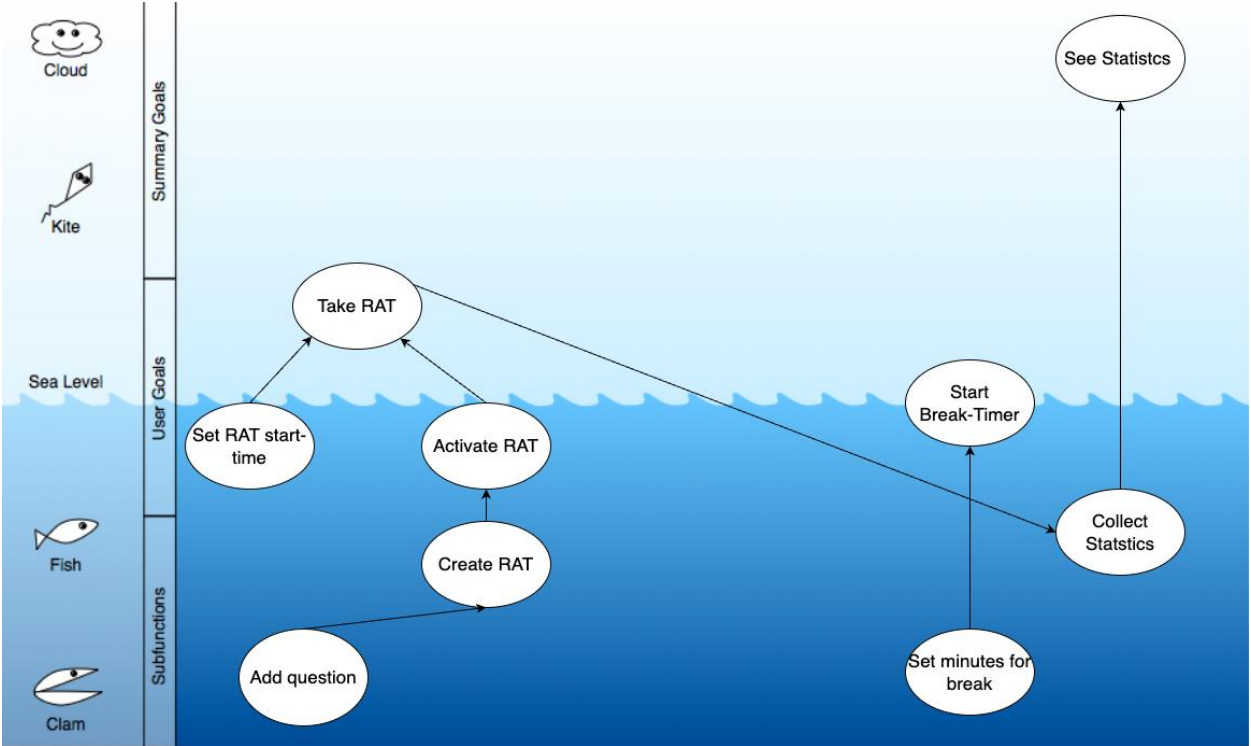
- Reduce paper sheets total amount for RATs by 95%.
 - Since there is now essentially no need for printing out the RATs we can reduce the amount of paper massively as there is little to no other use of paper in class.
- No more than 1 hour of class time should be spent answering questions about the system the first day it is in use.
- Reduce the time spent on taking the RAT by 20%.
 - This is the time wasted from it being on paper. Like waiting for the paper to be delivered to your table etc.
- Our system should work on most browsers, at least 98% of the students should be able to use their own device.

Stakeholders

Stakeholders	Value & interests
Educational Institution	The educational institution, or the university, will have an interest in this project in the form of a system that makes the running of the course easier. If this system works well for this course, it could also be used or/and adapted for other courses. They can find the project to finish and implement in their institution. The university will also be the ones funding the project, which is their value to us.
Student	The students will get a better experience from the course with our system. They are valuable for testing the system, giving us important feedback on the areas of the system they will interact with.
Developer	The developers will oversee building the entire system from implementation to testing.

Teachers/ Teaching Assistants	The teachers and teaching assistants are important for feedback and testing in the same way as the students, as they provide valuable info about the use of the parts of the system meant for them (like view RATs results).
-------------------------------------	--

Use Case Sea-Level Overview



Action	Level
Add question	Clam
Create RAT	Fish
Activate RAT	Fish
Take RAT	Sea
Set minutes for break	Clam
Start break-timer	Sea
Collect Statistics	Fish
See Statistics	Cloud

Use Cases in Table Form

Use Case 1:

ID and Name	T-1 Take RAT (Readiness Assurance Test)
Primary Actor	Student
Secondary Actor	TAs (teaching assistants)
Description	The student requests to take the RAT on the webpage and is sent to the right Google Form. The student then completes the RAT by filling out the form, and then the results are sent to the python program to be processed.
Trigger	The student clicks on the RAT link
Preconditions	PRE-1. Student needs a device which can open a website PRE-2. Student needs a student email

Postconditions:	POST-1 Result/answers is sent to server for processing.
Normal Flow	<p>1.0 Take RAT</p> <ol style="list-style-type: none"> 1. Student enters the webpage 2. Student clicks the RAT link 3. The student is sent to the Google Form and takes the RAT 4. The server with the python code gathers and handles the data from the google form 5. The server handles the data and makes the statistics available for the Teachers
Alternative Flows	<p>1.1 RAT is not open</p> <ol style="list-style-type: none"> 1. The student enters the webpage 2. Student clicks the RAT link 3. The embedded RAT does not show, the page only displays the title. 4. The student requests the TA or the teacher 5. If the RAT is supposed to be open, the teacher enters the teachers view to activate the RAT.
Exceptions	<p>1.2 E1 Student haven't taken the RAT before it was closed</p> <ol style="list-style-type: none"> 1. They contact the teacher which can re-active the RAT if necessary. The statistics are then updated as normal.

Priority	High
Frequency of use	One time each week by each student.
Other Information	-
Assumptions	-

Use Case 2:

ID and Name	T-2 Start break timer
Primary Actor	Student
Secondary Actor	-
Description	One student in the group pushes the pause button, starting a pause timer so the group can have a break. The button-push is registered by the server which starts the countdown. At the end of the countdown there is an alarm, and the timer is reset.
Trigger	Student pushes the pause button from their

	device.
Preconditions	PRE-1. Student needs a device which can open a website
Postconditions:	POST-1 Countdown finishes and restarts
Normal Flow	<p>2.0 Take a break</p> <ol style="list-style-type: none"> 1. Student enters the webpage 2. Student presses the pause button 3. The countdown starts 4. The countdown finishes 5. There is an alarm 6. Timer is reset
Alternative Flows	<p>2.1 Break is ended earlier</p> <ol style="list-style-type: none"> 1. The break is started 2. Student ends the timer before the set time 3. No alarm rings 4. Timer is reset
Exceptions	2.1 E2: Page closes/refreshes

	1. The timer is reset and you have to set the value anew.
Priority	Low
Frequency of use	About 2 times a week for each team.
Other Information	-
Assumptions	-

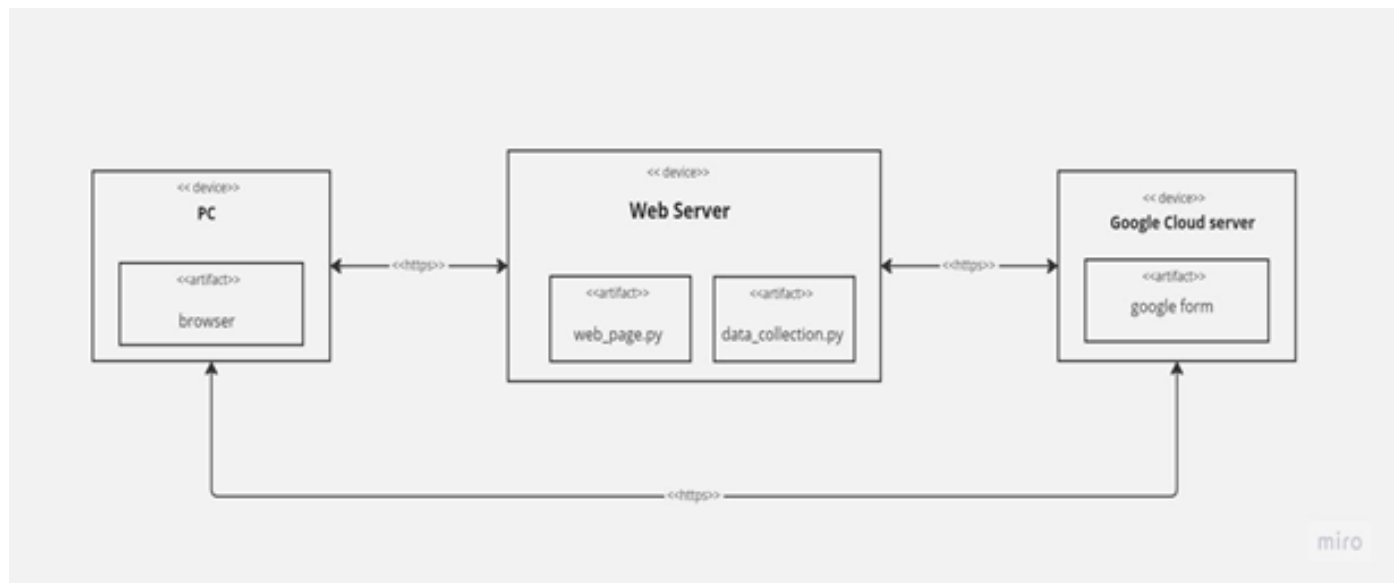
Use Case 3:

ID and Name	T-3 See Statistics
-------------	--------------------

Primary Actor	TAs and teachers
Secondary Actor	The university
Description	The TAs want to see the result and statistics of the RATs.
Trigger	TA requests statistics
Preconditions	PRE-1 At least 1 RAT must have been finished for there to be any statistics
Postconditions:	POST-1 Statistics are presented on the webpage
Normal Flow	<p>3.0 Request statistics</p> <ol style="list-style-type: none"> 1. After RAT, the result of the google form is sent to be handled 2. The data is sorted 3. The TA requests statistics 4. The statistics are collected from the python script that handles it and presented on the webpage for the TA to see

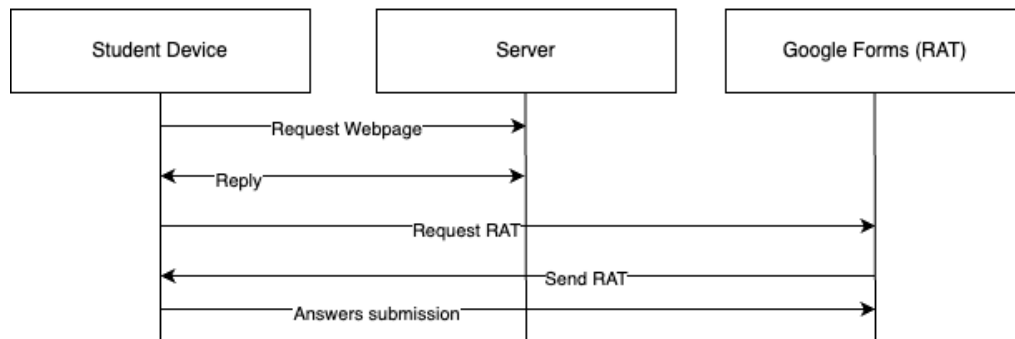
Alternative Flows	<p>3.1 Student needs to take RAT on paper</p> <ol style="list-style-type: none"> 1. The student contacts the teacher/TA to let them know that they cannot do it digitally 2. The teacher hands them a paper version 3. The student answers and gives the paper to the teacher 4. The teacher enters the data into the google form to update the statistics
Exceptions	<p>3.2 E3 No RAT data</p> <ol style="list-style-type: none"> 1. The statistics is simply blank
Priority	High
Frequency of use	At least once a week, after each RAT
Other Information	-
Assumptions	-

Deployment Diagram

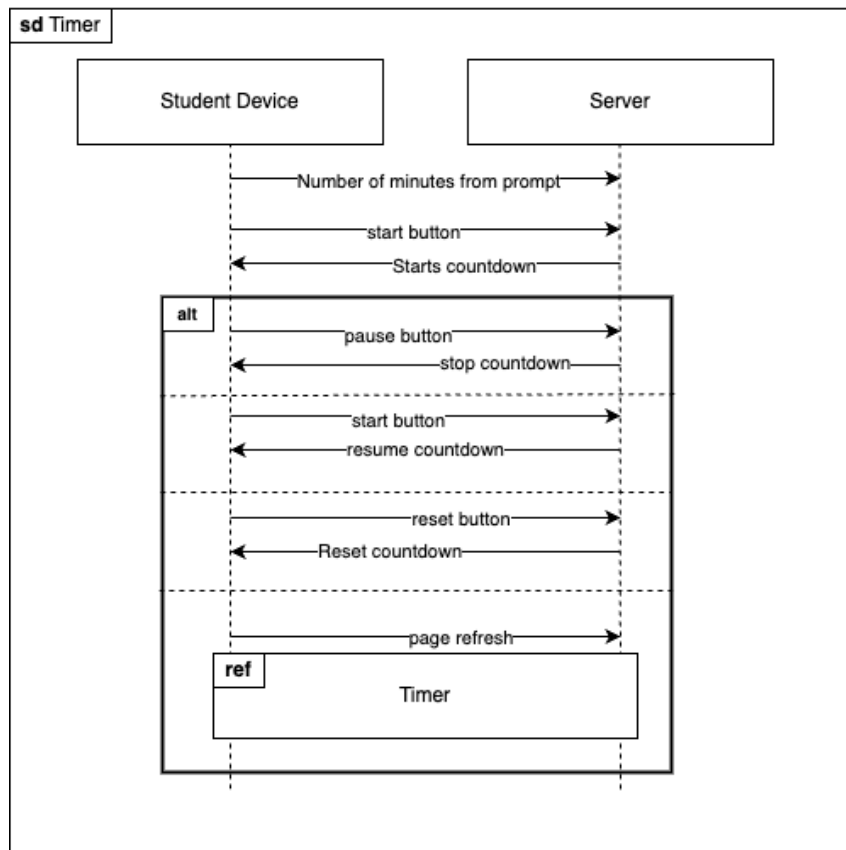


Sequence Diagrams

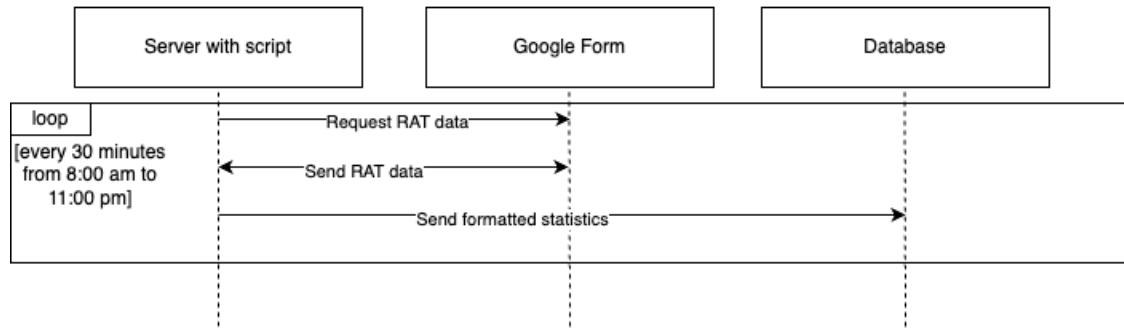
Taking RAT



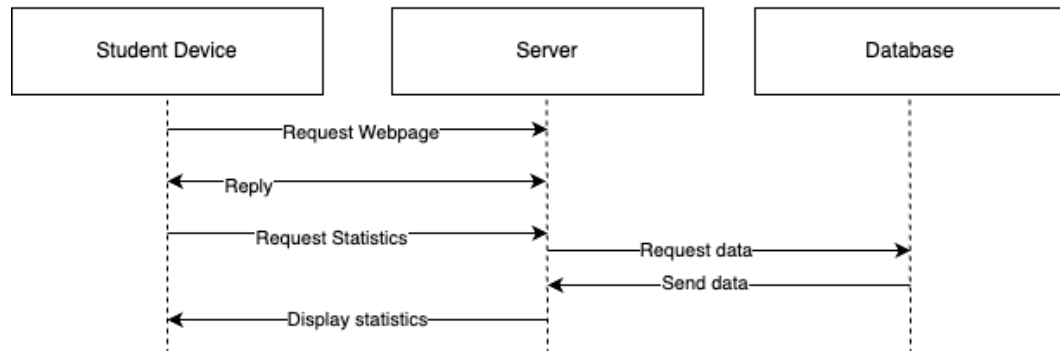
Starting Break Timer



Collecting statistics

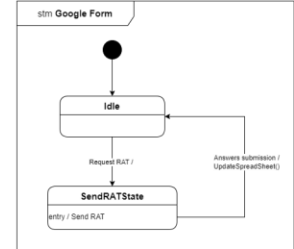
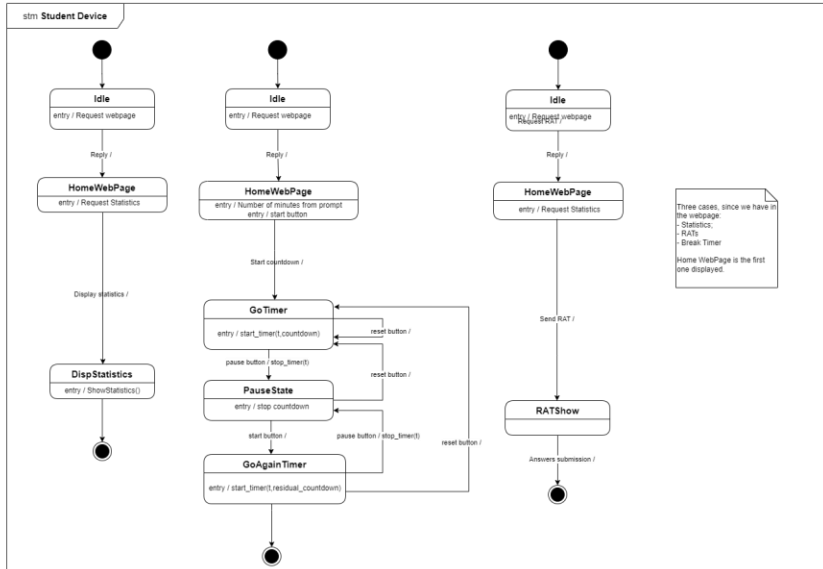
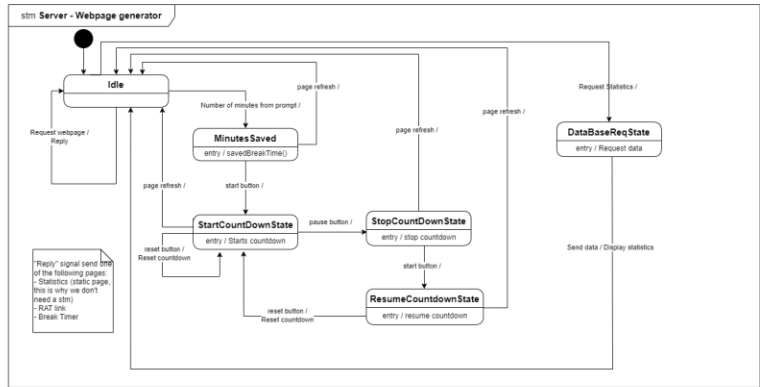
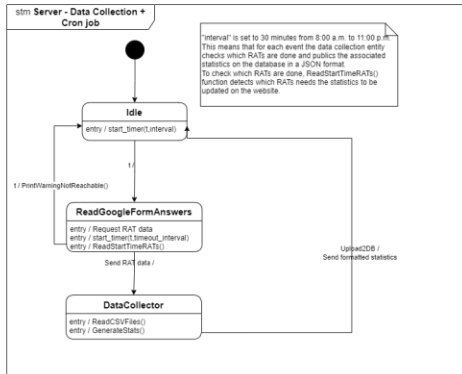


See Statistics



State Machine Diagrams

STM



Implementation Comments

Git Repository: https://github.com/tan90vir/TTM4115-group_13

The implementation of the system

Mostly everything we wanted to implement has been implemented, with some limits and not everything working perfectly. We have a website showing statistics, the break timer, and with the RAT as an embedded google form. The website is lacking in login functionality, meaning the functionality only meant for teachers and TAs are now accessible for everyone. Having a whole user system was a bit too complicated and time consuming for the time we had for the project, especially considering we are only 4 people.

- **The website (Rats, Statistics, Break-Timer)**

For the webpage we have used the Django framework, and the firebase real-time database for the collection of the statistics and transferring them in between.

The website is the main feature of our system. This is where both students and teachers will find all the functionality. The website has several pages which are intuitive to navigate in between.

For the teachers view the teachers can activate and deactivate RATs. We have 3 RATs as an example but this can rather easily be extended. The teachers also has to press a button to stamp the start time of the RAT. This is to be able to find the average time the students spend solving the RAT, as we only get the delivery time from the Google Form. This was difficult to have in a single button, but we would have preferred it if this time stamp was included in the “activate” button for ease of the user.

When the teacher presses the activate button the RAT will be available for the students. Until then the RAT will be hidden, so they cannot start the RAT before the teacher makes it visible.

There is a dropdown menu listing all the RATs. If a student enters a RAT before it is opened there is simply nothing on the page. When the RAT is activated they will see an embedded Google Form they can fill in.

There is also an overview of the statistics. Here the collected statistics are displayed in a spreadsheet, with the overview over which RAT the statistics belong to.

The break timer will give you a prompt with how many minutes you want your break to last. Then there is a clock showing the remaining time (a countdown), that can be paused and reset. We did not manage to implement an alarm sound which we wanted to, which means the student would have to keep an eye out themselves for how long is left of the break. This is not ideal and is something we would implement if we had more time. If

one refreshes the page the prompt will be given again, which is unlucky if one refreshes the page on accident without being aware.

- **Statistics collection and handling**

The collection and handling of the data from the Google Form is done with a python script.

The script gets the data from the Google Form using google sheets API to get the table as a CSV. Then the data is formatted into sensible statistics, in our case:

Number of participants, the average score (in %), the time spent on the RAT, and the average for each question. This is then uploaded to a real-time database which the webpage collects the information from and then displays.

Please note that there is some bug with the calculation of the time spent where we get negative numbers.

The python script is called from a bash script once every RAT.

What we are proud of

We are proud that we managed to get all our separate parts to work together, and that we managed to find a way to hide the RATs until the teachers activate them. This was a bit tricky and we are proud and happy that we managed this functionality. As well as the interaction between the teacher setting the timestamp, this getting uploaded to a txt file which is then read by the script to find the average time taking the RAT. We are in general very proud of the product as a whole, even if it is not perfect.