

# A Student Friend ELO



## Delivery Document

Group DEV-A5-1

Version 3.12

Client: Floor W. / Jan-Chris H.

## TABLE OF CONTENTS

PROJECT PLAN.....	4
Project Objectives.....	4
Agreement with Client.....	5
Project Requirements & Timeline.....	5
Group Members.....	7
PLAN OF APPROACH .....	8
SYSTEM PORTFOLIO.....	8
Functional Design.....	8
Technical Design .....	8
Vue.js .....	8
Vuex .....	9
Data fetching and error handling .....	11
Database .....	12
API description.....	13
Login page.....	13
Register page .....	14
mySaxion page (home page) .....	14
exam-schedule.....	15
Find teacher .....	15
email .....	15
exam-score.....	15
subject.....	15
news.....	16
schedule .....	16
Course .....	16
Download file .....	16
Peer study-assignment data .....	16
Peer study (assignments submitted) .....	17
Introduction page .....	17
Submit point .....	17
Video .....	17
Instruction.....	17

Assignment .....	18
Lectures(online class) .....	18
Lectures-archive.....	18
Implementation Report .....	19
Sprint 1.....	19
Sprint 2:.....	21
Sprint 3.....	26
spider map .....	26
peer study .....	27
Quiz & Progres bar .....	30
Discussion .....	32
Lectures.....	33
PROJECT PORTFOLIO.....	34
Code of Conduct .....	34
Scum-artefacts.....	34
Sprint 1.....	34
Scrum retrospective.....	34
Sprint's backlog.....	35
Burndown chart .....	36
Scrum related reports.....	36
Sprint 2.....	37
Scrum retrospective.....	37
Sprint's backlog.....	37
Burndown chart .....	39
Scrum related reports.....	39
Timesheet overview.....	40
Plan for coming sprint.....	40
Sprint 3.....	40
Scrum retrospective.....	41
Sprint's backlog.....	41
Burndown chart .....	45
Scrum related reports.....	46
Timesheet overview.....	46

Proof of Concept .....	46
ICT Card: Survey .....	47
Meetings .....	51
Questions for client .....	51
Client Meetings.....	51
ICT Research Methods .....	53
ICT Card: Ideation .....	54
Customer Journey map.....	55
Activity map .....	55
BPMN .....	56
Wireframes .....	56
1 <sup>st</sup> Concept wireframe .....	56
2 <sup>nd</sup> Concept wireframe.....	57
3 <sup>rd</sup> Concept wireframe .....	58
Final concept wireframe .....	61
Individual Reflections.....	62
Declaration of Competences .....	63
Yang.....	63
Mykhailo .....	63
Resources.....	64
Levels of competences.....	64
Todo applications.....	64
Som Model English .....	64
Final Survey link: .....	64
GitLab Repository .....	64

## PROJECT PLAN

### Project Objectives

The project objective is to improve the student learning experience. This aim is to create an easy to navigate student portal system where all relevant information is quickly available including course module materials.

## Agreement with Client

- A new better blackboard for Saxion where both the homepage and course page will be easier and more efficient to use for students. The homepage should include all the important features the student needs the most and should be easy to navigate through. Important features: Easy access to subjects, news, grades, new implementation of the competences, schedule and exams.
- For the course page it should benefit the students, rather than what is going on in the current situation. An important feature should be that the student can discuss their work peer-to-peer so that student can learn and help each other through their school journey.  
Important features: better navigation, peer-to-peer review.

The role of the customer is to let us know which features are needed for this project and to help us along the way. When we create wireframes or ideas the customer should give us feedback on mentioned ideas and help us achieve a better project.

## Project Requirements & Timeline

//Requirements of project and deadlines or scrum meetings...

The development time of the entire project is 8 weeks. Each development sprint is 2 weeks. At the end of sprint0, we will submit our final PoA. At the end of sprint1, we will submit the final design document and the completed part of the code. At the end of sprint2, we will submit our code. Which completed most of the functions. At the end of sprint3, we will submit the final document and the code of the entire project.

### **Sprint3 requirements**

Id	Description	User/System /Business requirement	Priority (1=important)	Moscow
B-01	The final product can improve Saxion's existing digital education environment according to existing students and the client.	Business	1	Must
U-01	The course page has navigation bar, and it can clearly reflect the course structure.	User	1	Must
U-02	a spider map to reflect the learning situation of students, and the data in it is calculated automatically through the course scores.	User	1	Must
U-03	The course page needs to have a video function, students can watch the video uploaded by the teacher.	User	1	Must
U-04	The course page requires an assignment submission point where students can submit their final assignments.	User	1	Must

U-05	The course page needs to have a quiz function. Students complete the quiz and get their scores immediately. If they fail to pass, they can retake.	User	1	Must
U-06	On the course page, the introduction interface needs to have a progress bar to reflect the student's learning progress.	User	1	Must
U-07	On the course page, students can join the online course and view the recording.	User	1	Must
U-08	On the course page, there is an introduction interface to provide students with the details of the course.	User	1	Must
U-09	Students can login and register.	User	2	Should
U-10	On the course page, there is a peer study interface, a platform where student can share code or homework ideas with other students. Students can learn from each other.	User	1	Must
U-11	In the peer study interface, students can download other student's works.	User	2	Should
U-12	Footer link, links to other websites of the school, such as the library.	User	3	Could
U-13	Discussion page, a built-in discussion platform, including two channels for discussing homework with classmates and discussing code issues with teachers.	User	1	Must
S-01	Connection between components, make learning orderly. the components in the course to be related to each other and set up in order.	System	1	Must
U-14	On the course page, there is an assignment interface to show all assignments, including questions, pictures, and difficulty.	User	1	Must
U-15	System failure information, student can get the system's error description in time when the server fails.	User	1	Must
U-16	Home page includes exam-schedule, schedule, Email, Find teacher, news, spider-map(in a pop-up window)	User	1	Must
U-17	Exam-schedule component use the calendar to clearly show the test date.	User	1	Must
U-18	Schedule component contains all learning activities of student.	User	3	Could
U-19	Email component lists the student's emails	User	1	Must

U-20	Find teacher component helps students to quickly get the teacher's email information.	User	3	Could
U-21	News component lists the latest news or announcements from the school.	User	3	Could
S-02	Use SQLite database.	System	1	Must
S-03	Use API to transfer data.	System	1	Must
S-04	Use vuex store to manage the state of all components of the application.	System	1	Must
S-05	Use vuetify instead of HTML CSS on the course page.	System	1	Must

### Group Members

- Mykhailo G - 470934
- Yang Cheng - 474340

## PLAN OF APPROACH

Can be viewed at this link or in the Git documentation folder

[https://saxion.sharepoint.com/:w/r/teams/o365-team005862/Gedeelde%20documenten/General/2.4-IT\\_Corp\\_Plan\\_of\\_Approach.docx?d=wc9dc24be792c41d0ab8f6ecc6eb121e&csf=1&web=1&e=baBrm4](https://saxion.sharepoint.com/:w/r/teams/o365-team005862/Gedeelde%20documenten/General/2.4-IT_Corp_Plan_of_Approach.docx?d=wc9dc24be792c41d0ab8f6ecc6eb121e&csf=1&web=1&e=baBrm4)

## SYSTEM PORTFOLIO

### Functional Design

Users can experience a new redesigned Blackboard home and module page. We took all of the most important things that students use every day and made them easily accessible and intuitive.

The homepage includes tiles of the primary information like courses, schedule, exam schedule, and more.

Modules now have improved structure and additional features. Peer review, quiz, progress bar, structured videos and more, are included in the updated module page.

Students can see their progress in a course on the progress bar. This is calculated on the basis of completed peer study assignments and quizzes.

Peer study: The purpose of peer study is to allow students to share their assignment with other students and learn from each other. In peer study, the completion of the assignment should be orderly. When the student completes the first assignment: uploading, downloading other people's work, and giving at least 1 comment, the assignment2 can be started.

Quiz: quiz is locked at the beginning, only after completing all assignments in the peer study, it will be unlocked. Quiz supports 4 type question: choose, multiple choose, text question and True false question. When the user completes the quiz, the system will compare the user's input with the correct answer and give the final score. If the score is less than 4 points, it will be judged as insufficient, and the user can retake. If the score is  $\geq 4$  points, it means the user Passed quiz, and completed all the study this week, the progress bar in the introduction page will move to the next week.

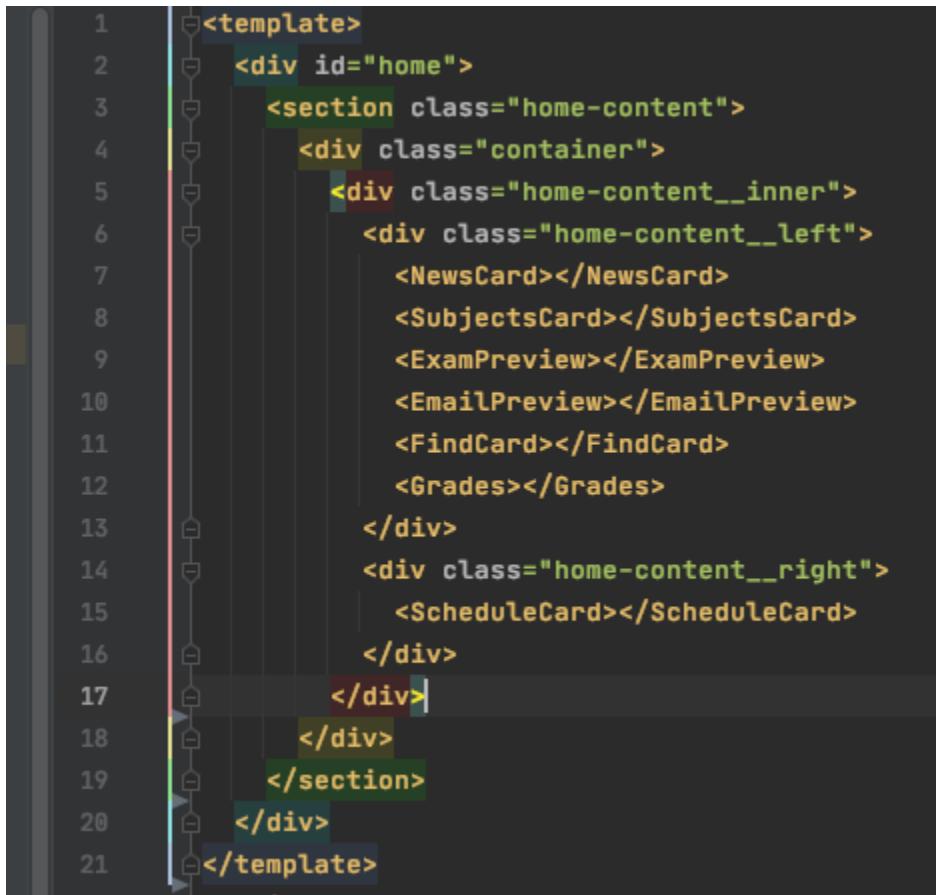
### Technical Design

#### Vue.js

The framework used in this project is [Vue.js](#). This we chose because it is very demanded in the market and it's something that we still had yet to learn.

The application is divided into small Vue components, so it is more scalable and maintainable.

An example of a file that uses multiple Vue components is shown below:



```
1 <template>
2   <div id="home">
3     <section class="home-content">
4       <div class="container">
5         <div class="home-content__inner">
6           <div class="home-content__left">
7             <NewsCard></NewsCard>
8             <SubjectsCard></SubjectsCard>
9             <ExamPreview></ExamPreview>
10            <EmailPreview></EmailPreview>
11            <FindCard></FindCard>
12            <Grades></Grades>
13          </div>
14          <div class="home-content__right">
15            <ScheduleCard></ScheduleCard>
16          </div>
17        </div>
18      </div>
19    </section>
20  </div>
21 </template>
```

The whole home page with lots of data is written in 20 lines of template code. This is the beauty of Vue.js

## VueX

We used [VueX](#) for our state management and passing common data between components.

VueX file is located in the src/store folder.

The state of the data is stored in the state object.

```

9   state: {
10    weekFinish: 0,
11    grades: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
12    course: {
13      introArticles: [],
14      submitAssignments: [],
15      peerStudy: {
16        finishedPeerStudy: 0,
17        assignments: [],
18        submittedAssignments: []
19      },
20      moduleAssignment: {
21        assignments: []
22      },
23      video: {
24        tasks: []
25      },
26      instruction: {
27        instructions: []
28      },
29      lectures: {
30        lecture: [],
31        archive: []
32      },
33      weekPages: []
34    }
35  },

```

Mutation object is used to set the data when it is retrieved and update it:

```

59   SET_SUBMIT_ASSIGNMENTS (state, submitAssignments) {
60     state.course.submitAssignments = submitAssignments
61   },
62   SET_PEER_STUDY_SUBMITTED_ASSIGNMENTS (state, submittedAssignments) {
63     state.course.peerStudy.submittedAssignments = submittedAssignments
64   },
65   UPDATE_PEER_STUDY_SUBMITTED_ASSIGNMENTS (state, { submittedAssignment, idx }) {
66     state.course.peerStudy.submittedAssignments[idx].downloaded = true
67   },

```

For this project we have 2 getters that calculate us whether the user is allowed to take the quiz and a peer study assignment. This data is calculated based on the state.

```

83   },
84   getters: {
85     allowedToDoQuiz: state => {
86       return state.course.peerStudy.assignments.length <= state.course.peerStudy.finishedPeerStudy
87     },
88     allowedToDoAssignment: state => (assignmentNumber) => {
89       return !(assignmentNumber <= state.course.peerStudy.finishedPeerStudy)
90     }
91   },

```

Action object is used to retrieve data from API calls. When the data is retrieved, we commit (update) the data in the store.

```
actions: {
  fetchSubmitAssignments ({ commit }, { weekId }) {
    axios
      .get('http://localhost:3000/api/submit-point', config: { params: { weekId } })
      .then(response => {
        commit('SET_SUBMIT_ASSIGNMENTS', response.data)
      })
  },
  fetchIntroArticles ({ commit }, { courseId }) {
    axios
      .get('http://localhost:3000/api/intro-page', config: { params: { courseId } })
      .then(response => {
        commit('SET_INTRO_ARTICLES', response.data)
      })
  },
}
```

We call actions function from a Vue component when it's mounted.

Vuex supports plugins. For this project we used a plugin that allows to save the data in the sessions. That way, the data stays the same when we reload the page.

```
166   plugins: [
167     createPersistedState( options: {
168       storage: window.sessionStorage,
169       reducer (state) {
170         return {
171           weekFinish: state.weekFinish,
172           course: state.course
173         }
174       }
175     }
176   )
177 ]
```

## Data fetching and error handling

To consume APIs we used [axios](#) because Vue.js supports it the most.

The data is fetched when the component is mounted. There are the following fields in the Vue data object: the data itself ('teachers', on the picture below), 'loading' (specifies when the data is loading) and 'errored' (specifies when an error occurred).

```

49   data: () => ({
50     teachers: [],
51     loading: true,
52     errored: false,
53   }),
54   methods: {},
55   mounted() {
56     axios
57       .get('http://localhost:3000/api/find-teacher')
58       .then(response => {
59         console.log(response)
60         this.teachers = response.data;
61       })
62       .catch(error => {
63         console.log(error)
64         this.errored = true
65       })
66       .finally( onFinally: () => this.loading = false)
67   }

```

When the data is loaded, we assign it to the necessary variable and then set the ‘loading’ variable to false.

If the backend is not available, the catch block is executed, errored variable is set to true and the user is shown a message that the service is not available.

Template example of a component (Find-card.vue):

```

19   <section v-if="errored">
20     <p class="pa-2">We're sorry, we're not able to retrieve this information at the moment, please try again
21     later</p>
22   </section>
23   <section v-else class="find-card__inner">
24     <div v-if="loading" class="pa-2">Loading...</div>
25     <div
26       v-else
27       v-for="item in teachers"
28       class="find-item"
29     >
30       <div class="find-item__image">
31         
32       </div>
33       <div class="find-item__contact-details">
34         <div class="find-item__text"><strong>Name:</strong> {{ item.teacher_name }}</div>
35         <div class="find-item__text"><strong>Email:</strong> {{ item.teacher_email }}</div>
36         <div class="find-item__text"><strong>Abbreviation:</strong> {{ item.teacher_abbreviation }}</div>
37       </div>
38     </div>
39   </section>

```

## Database

We use [SQLite](#) because it is easy to install and because it can handle the size of the project. In addition, SQLite does not require many resources to run.

When the server starts, SQLite checks the tables, drops them, creates new ones, and inserts data.

```

16
17     //Intro page
18     await db.exec( sql: "DROP TABLE IF EXISTS course_intro ; CREATE TABLE IF NOT EXISTS course_intro (\n" +
19         "\tid INT,\n" +
20         "\tcourse_id INT,\n" +
21         "\ttitle VARCHAR(50),\n" +
22         "\timage VARCHAR(50),\n" +
23         "\ttext VARCHAR(50)\n" +
24         ");\n" +
25         "insert into course_intro (id, course_id, title, image, text) values (1, 23 , 'Learning progress', '', '');\n" +
26         "insert into course_intro (id, course_id, title, image, text) values (2, 23 , 'Introduction', '', 'Hello students (that participated in\n' +
27         "insert into course_intro (id, course_id, title, image, text) values (3, 23 , 'Course Manual', 'course-manual.png', '');\n"
28     );
29

```

Later, when we have a DB object, we use it to retrieve the data.

```

506         let courseId = req.query.courseId;
507         if (!courseId) return res.json( body: [] )
508         let result = await db.all(`SELECT * FROM lectures_lecture WHERE course_id = ?`, courseId);
509         res.json(result);

```

### API description

[Express.js](#) framework is used to create API routes. The description of the API routes you can find in the tables below.

Each of the routes has a setTimeout() function. This is used to delay data retrieval so that the front end displays the loading functionality.

```

502     // lectures-lecture
503     router.get( path: "/api/lectures-lecture", handlers: async function (req: Request<P, ResBody, ReqBody, ReqQuery, Locals>)
504     try {
505         setTimeout( handler: async function () {
506             let courseId = req.query.courseId;
507             if (!courseId) return res.json( body: [] )
508             let result = await db.all(`SELECT * FROM lectures_lecture WHERE course_id = ?`, courseId);
509             res.json(result);
510         }, timeout: 500);
511     } catch (e) {
512         res.json(e);
513     }
514 );

```

### Login page

Functionality	User Login
URL	<a href="http://localhost:3000/api/login">http://localhost:3000/api/login</a>
Method	POST
Param info	<ul style="list-style-type: none"> <li>- user_id      (Auto-growth in database)</li> <li>- userName    (User's login name)</li> <li>- passWord    (User's login password)</li> </ul>
Return Msg (Json)	//successful login { Status: 200 //Jump to home page }

	<pre>// login failed { Status code: 500 Message: "Wrong username and password, please try again" }</pre>
--	--

*Register page*

Functionality	User Register
URL	<a href="http://localhost:3000/api/register">http://localhost:3000/api/register</a>
Method	POST
Param info	<ul style="list-style-type: none"> <li>- userName (User's name/email for login)</li> <li>- passWord (User's password for login)</li> <li>- passWordConfirm (enters the same password to confirm)</li> </ul>
Return Msg (status code)	<pre>//successful register { Status code: 200; //back to login page } //Wrong format { Message: "Incorrect username or password format." }</pre>

*mySaxion page (home page)*

Functionality	After login, the user will see home page
URL	<a href="http://localhost:8080">http://localhost:8080</a>
Method	GET
Param info	-
Return	Return map; (map include course list, schedule list, news list, grades list, email list, message list and chat list)

*exam-schedule*

Functionality	Get all exam schedule from database &display on homepage
URL	<a href="http://localhost:3000/api/exam-schedule">http://localhost:3000/api/exam-schedule</a>
Method	GET
Param info	-
Return Msg (Json)	Return exam-schedule list.

*Find teacher*

Functionality	Get all teacher from database &display on homepage
URL	<a href="http://localhost:3000/api/find-teacher">http://localhost:3000/api/find-teacher</a>
Method	GET
Param info	-
Return Msg (Json)	Return teacher list.

*email*

Functionality	Get all email from database &display on homepage
URL	<a href="http://localhost:3000/api/email-preview">http://localhost:3000/api/email-preview</a>
Method	GET
Param info	-
Return Msg (Json)	Return email list.

*exam-score*

Functionality	Get all exam grade from database &display on homepage
URL	<a href="http://localhost:3000/api/exam-score">http://localhost:3000/api/exam-score</a>
Method	GET
Param info	-
Return Msg (Json)	Return exam-score list.

*subject*

Functionality	Get all subject from database &display on homepage
URL	<a href="http://localhost:3000/api/subjects">http://localhost:3000/api/subjects</a>
Method	GET
Param info	-
Return Msg (Json)	Return subject list.

*news*

Functionality	Get all news from database & display on homepage
URL	<a href="http://localhost:3000/api/news">http://localhost:3000/api/news</a>
Method	GET
Param info	-
Return Msg (Json)	Return news list.

*schedule*

Functionality	Get all schedule from database & display on homepage
URL	<a href="http://localhost:3000/api/schedule">http://localhost:3000/api/schedule</a>
Method	GET
Param info	-
Return Msg (Json)	Return schedule list.

*Course*

Functionality	Enter a course(introduction to programming)
URL	<a href="http://localhost:8080/course">http://localhost:8080/course</a>
Method	GET
Param info	-
Return Msg (Json)	Return course page.

*Download file*

Functionality	Example of download other student's assignment
URL	<a href="http://localhost:3000/api/download">http://localhost:3000/api/download</a>
Method	POST
Param info	-
Return Msg (Json)	json data stream

*Peer study-assignment data*

Functionality	Get the assignment data in peer study page
URL	<a href="http://localhost:3000/api/peer-study">http://localhost:3000/api/peer-study</a>
Method	GET
Param info	week_id: int (Used to distinguish the peer study of each week)
Return Msg (Json)	Assignment list

*Peer study (assignments submitted)*

Functionality	Get the information about other student's work, name and submission date.
URL	<a href="http://localhost:3000/api/peer-study-submitted-assignments">http://localhost:3000/api/peer-study-submitted-assignments</a>
Method	GET
Param info	week_id: int (Used to distinguish the peer study (submitted) of each week)
Return Msg (Json)	Name and submission date

*Introduction page*

Functionality	Get the course introduction data from database
URL	<a href="http://localhost:3000/api/intro-page">http://localhost:3000/api/intro-page</a>
Method	GET
Param info	course_id : Int (Used to distinguish introduction in all courses)
Return Msg (Json)	A list, each element contains title, image path, and text.

*Submit point*

Functionality	Get the submit page data from database
URL	<a href="http://localhost:3000/api/submit-point">http://localhost:3000/api/submit-point</a>
Method	GET
Param info	week_id: int (Used to distinguish the submit point of each week)
Return Msg (Json)	A list, description of the submission points for different assignment

*Video*

Functionality	Get the video page data from database
URL	<a href="http://localhost:3000/api/video">http://localhost:3000/api/video</a>
Method	GET
Param info	week_id: int (Used to distinguish the video of each week)
Return Msg (Json)	A list, description of the video content. (title,subtitle)

*Instruction*

Functionality	Get the instruction page data from database
URL	<a href="http://localhost:3000/api/instructions">http://localhost:3000/api/instructions</a>

Method	GET
Param info	week_id: int (Used to distinguish the instruction of each week)
Return Msg (Json)	A list, description of each instruction (title,subtitle,text)

#### *Assignment*

Functionality	Get the assignment page data from database
URL	<a href="http://localhost:3000/api/assignment">http://localhost:3000/api/assignment</a>
Method	GET
Param info	week_id: int (Used to distinguish the assignment of each week)
Return Msg (Json)	A list, each assignment includes a title, subtitle, rating(difficulty), and path to the image.

#### *Lectures(online class)*

Functionality	Get the data in the lecture page, including the title and introduction of the lecture
URL	<a href="http://localhost:3000/api/lectures-lecture">http://localhost:3000/api/lectures-lecture</a>
Method	GET
Param info	course_id : Int (Used to distinguish online class in all courses)
Return Msg (Json)	title and introduction of each lecture.

#### *Lectures-archive*

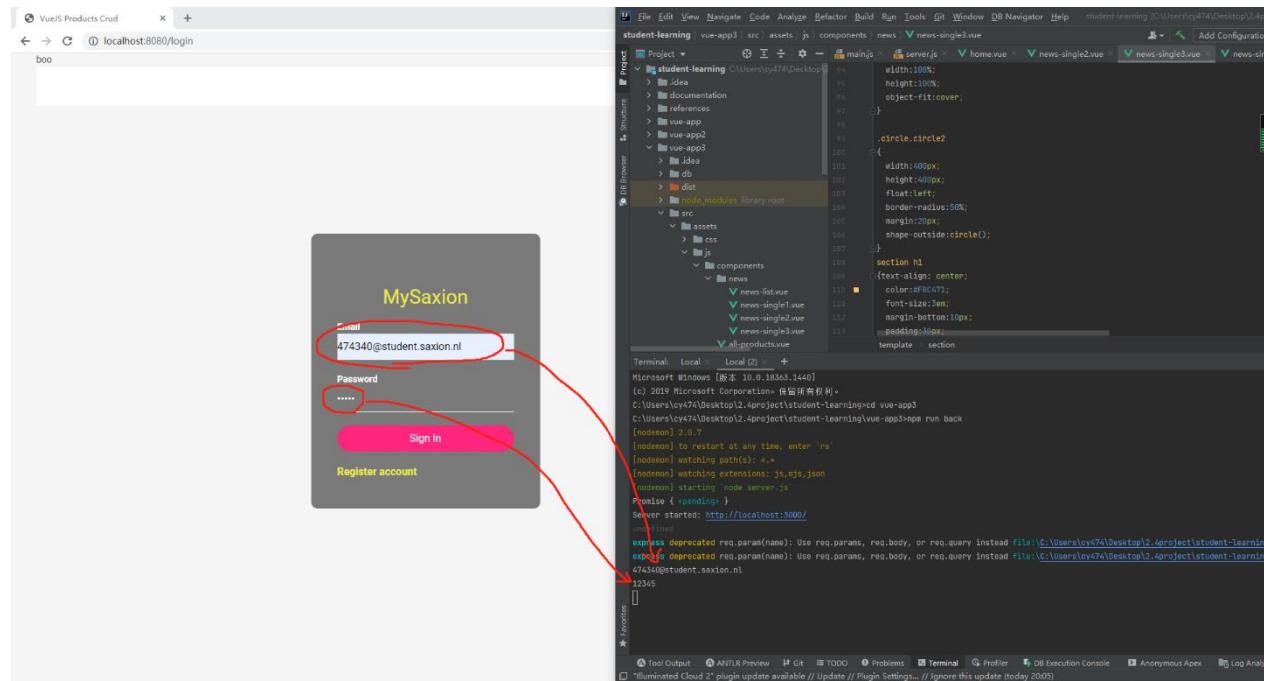
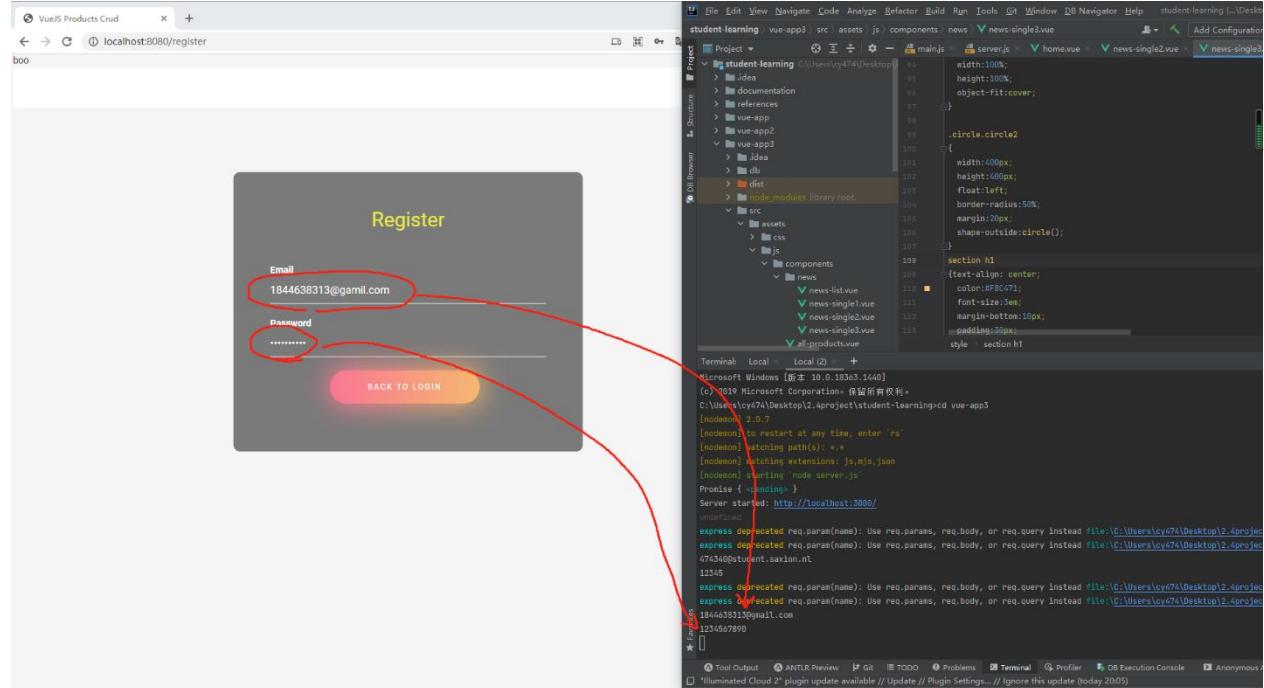
Functionality	Get the data of the recording page, including week number, specific start, end time, and duration.
URL	<a href="http://localhost:3000/api/lectures-archive">http://localhost:3000/api/lectures-archive</a>
Method	GET
Param info	course_id : Int (Used to distinguish recording in all courses)
Return Msg (Json)	week number, specific start, end time, and duration of each recording.

# Implementation Report

In sprint1. We have implemented login page, register page, home page and news page.

## Sprint 1

Every page has been tested, login and register can send user information to the backend through fetch API. Home page and news page can run HTML, CSS and Js normally.



04-05-2021 Corona-update: Request self-tests from May 5



From Wednesday May 5 it is possible for students and employees to request self-tests. These self-tests are free and you can request free self-tests through the national portal: [www.zelftestonderwijs.nl](http://www.zelftestonderwijs.nl). Use your Saxion account to log in to the portal. The tests will be sent free of charge to the address you provided.

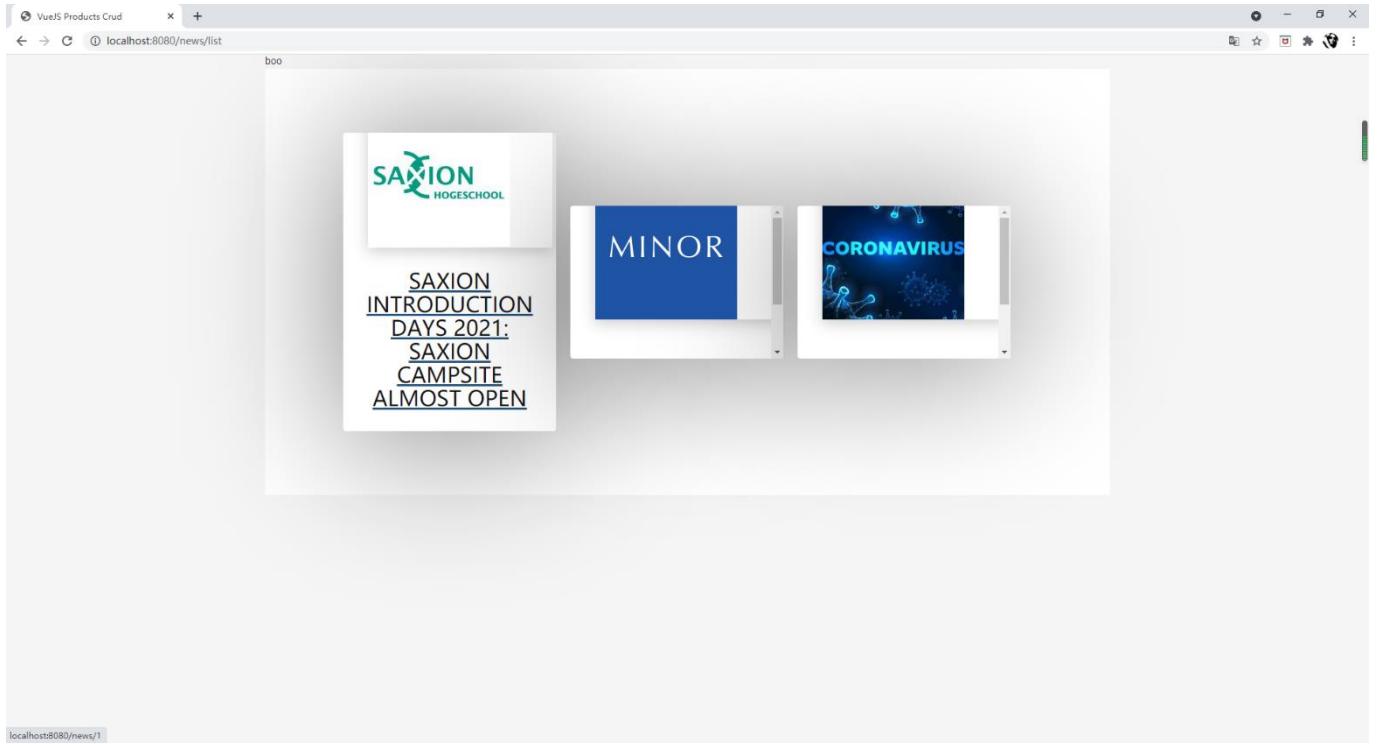
### What is self-testing?

As the name implies, self-testing is a test that you can perform yourself and which gives you a result within 30 minutes. By doing a self-test you help us keep everyone safe in our Saxion-buildings. The self-tests make it easier, and quicker, to trace any possible corona infections. This way we can stop the virus spreading unnoticed. This is why we think it is important to facilitate self-testing. Of course we still have to follow all corona safety measures, e.g. keeping 1.5 metres distance, wearing face masks outside class rooms, regularly washing hands, and staying at home in the event of symptoms. Self-testing is an extra measure in addition to the existing corona guidelines already in place at our locations.

### Self-testing is voluntary

The use of self-tests is voluntary. You do not need to show a negative result from a self-test to get access to Saxion buildings, and as mentioned earlier, all other corona guidelines remain in place.

### Corona-related symptoms? Go to the GGD for a test

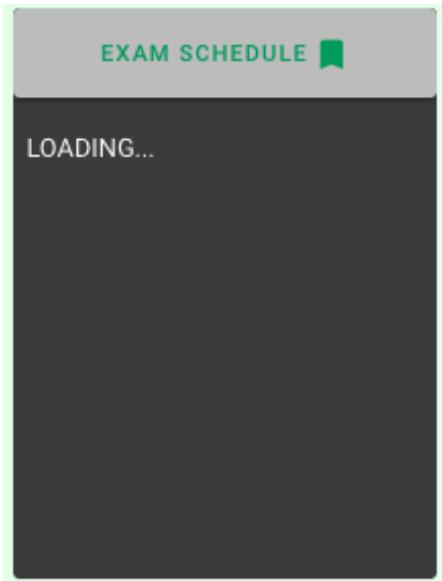


## Sprint 2:

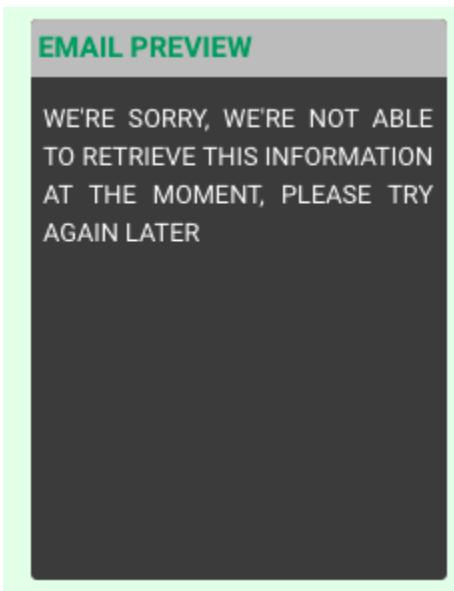
The home page has been improved in terms of code and design.

All tiles are placed in Vue components and connected to the API routes.

Most components have a feature that when the data is loading it will display "loading..."



The following information is displayed if a particular API cannot be reached, or an error occurred while retrieving data.



Module page is slightly different from what we have on wireframes, which is approved with the customer. Users have a menu for a particular week or other pages on top of the course page. Every page has a side menu with content related to that page.

The intro of a module looks like this:

The screenshot shows a Blackboard course page for 'Introduction'. At the top, there's a green header bar with the Saxon logo and navigation links for 'INTRO', 'WEEK 1', 'WEEK 2', 'WEEK 3', 'DISCUSSION', and 'EXTRA'. Below the header, a sub-header 'INTRODUCTION' is shown with a question mark icon. The main content area has a title 'Introduction' and a detailed text message from the teacher. Below this, a section titled 'Course Manual' contains a table for '1.1: Introduction to Programming' with columns for location, course year, quarter, credits, responsible teacher, exam code, exam type, and lessons per week. It also lists HBO-I competencies and specific learning goals and assessment details.

1.1: Introduction to Programming	
Location	Deventer
Course year	1
Quarter	1
Credits	5 EC
Responsible teacher	Tristan Pothoven
Exam code	T.S2328
Exam type	Digital (V/D)
Lessons per week	135 minutes 4-5 hrs seminar 6-7 hrs self study
<b>HBO-I competences</b>	
ANA ADV ONT REA M&C	
GI	TO ONG EKH MAN
OP	OP LPK ODK OPL
IN	PL END OWK PRF
SW	1 DI PAR COM SAM
HI	

**Learning goals**

Competence	Learning goal
SWREAH	You are able to create a properly readable and functioning Java program, given a certain specification.
SWREAH	You are able to define and use custom created data types.
SWREAH	You are able to apply the concept of methods to structure your code.
SWREAH	You are able to execute certain pieces of code multiple times using loops.
SWREAH	You are capable of storing multiple values in the same variable using Lists (including the usage of arrays and understanding the difference between lists and arrays).
SWREAH	You are able to allow or prevent the execution of pieces of code using conditions.

**Assessment**

During the test you will be asked to create three small Java programs. You will take the test on your own laptop, handing it in via Blackboard. The test will be made available at a specific time, after which you have 135 minutes to hand it in.

Week 1 page instruction looks as follows:

 INSTRUCTION ASSIGNMENT PEER STUDY VIDEO SUBMIT QUIZ**Discord**

Hello students, To assist you on your journey on learning how to program, we've created some additional help for you in the form of a Discord server. Please log in to this server to ask for assistance. So if you have questions, feel free to post them on the server. Usually there should be someone online to help out...

**How to install the Java Development Kit and IntelliJ IDEA Community Edition**

Set up system

During the first lecture we will install the required software together, but in case you are wondering we have created a step-by-step guide on how to install the software needed for this course. Head over to <https://www.oracle.com/java/technologies/javase-downloads.html> and go download the Java SE 11 (LTS) installer for your specific operating system. You'll need to create an Oracle account and sign in to be able to download the installer. Head over to <https://www.jetbrains.com/idea/download> and get IntelliJ IDEA Community Edition for your specific operating system. Install the Java SE 11 development kit downloaded from step 1. Unfortunately, there is no "easy" way to check whether or not the installation was a success. If the installer exited without errors, you should assume everything is up and running. If your installer crashes, please contact your teacher. Install IntelliJ IDEA Community Edition downloaded from step 2. Download the DemoProject.zip file (below) and extract it to somewhere you can easily find it. Open IntelliJ and select open and select the folder you extracted from the zipfile as discussed in step 5. Head over to "File: Project" (top left), open up "Demo1", open up "src" before double-clicking on "Application". You might need to wait before IntelliJ has indexed the Java Development Kit. If you see a progressbar in the bottom, wait a moment! Be patient. If everything worked out correctly, you should a green triangle that you can now click to run your program. If you can do so, your installation was successful! Just to help out, we have created a video that shows all these steps and explains a little about what happens. Have a look at <https://youtu.be/8Dp9jP56b4U>.

**Sandbox project**

Download the Sandbox project here. A Sandbox project does not contain exercises, but should be considered a playground for you to experiment upon!

Google it yourself!



Each week's assignment page has the following structure:

 INSTRUCTION ASSIGNMENT PEER STUDY VIDEO SUBMIT QUIZ

### assignment 1

level 1



Write a program, that greets you whenever you run it. For example:

```
Main
"E:\intelliJ\IntelliJ IDEA Community
Hello, Tristan!
Process finished with exit code 0
```

### assignment 2

level 2



Write a program that prompts the user for a name. Then greet the person with that name. For example:

```
Main
"E:\intelliJ\IntelliJ IDEA Community Edition
Enter username:
peter
Username is: peter
Process finished with exit code 0
```

### assignment 3

level 3



## Sprint 3

### spider map

localhost:8080

SAXION

NEWS

SAXION INTRODUCTION DAYS  
2021: SAXION CAMPSTE  
ALMOST OPEN  
READ ARTICLE

MAKE UP YOUR MINOR! FIND  
OUT MORE ABOUT OUR OFFER  
NOW  
READ ARTICLE

EMAIL PREVIEW

2021-06-09 21:28  
SENDER: DICK HEIJINK  
TITLE: PROJECT PASSED

2021-06-09 21:28  
SENDER: DICK HEIJINK  
TITLE: SUBMISSION RECEIVED

2021-06-09 21:28  
SENDER: FLOOR WEIJMA  
TITLE: GOOD JOB TEAM!

2021-06-09 21:28  
SENDER: DICK HEIJINK  
TITLE: SUBMISSION RECEIVED

2021-06-09 21:28

SUBJECTS

Q1 | Q2 | Q3 | Q4

INTRODUCTION TO PROGRAMMING  
IT AND LAW  
COMPIERS AND OPERATING SYSTEM  
DEVELOPMENT TOOLS

EXAM SCHEDULE

2021-06-09 21:28 HELLO WORLD

2021-06-09 21:28 HBO-ICT PROJECT

2021-06-09 21:28 WEB DEVELOPMENT

2021-06-09 21:28 DEV TOOLS

SCHEDULE

2021-06-09 21:28 C++  
2021-06-09 21:28 CURIOSITY  
2021-06-09 21:28 CONCURRENCY  
2021-06-09 21:28 IT AND LAW

FIND

TEACHER NAME

NAME: DICK HEIJINK  
EMAIL: J.D.HEIJINK@SAXION.NL  
ABBREVIATION: JDH

NAME: FLOOR WEIJMAN  
EMAIL: F.R.WEIJMAN@SAXION.NL  
ABBREVIATION: FRW

NAME: ROGIER HOMMELSEN  
EMAIL: R.HOMMELSEN@SAXION.NL

GRADES

YEAR 1 | YEAR 2 | YEAR 3 | YEAR 4

DEV TOOLS 9

HELLO WORLD 10

WEB APPLICATIONS 8

C++ 10

click cobweb button

localhost:8080

SAXION

NEWS

SAXION INTRODUCTION DAYS  
2021: SAXION CAMPSTE  
ALMOST OPEN  
READ ARTICLE

MAKE UP YOUR MINOR! FIND  
OUT MORE ABOUT OUR OFFER  
NOW  
READ ARTICLE

EMAIL PREVIEW

2021-06-09 21:28  
SENDER: DICK HEIJINK  
TITLE: PROJECT PASSED

2021-06-09 21:28  
SENDER: DICK HEIJINK  
TITLE: SUBMISSION RECEIVED

2021-06-09 21:28  
SENDER: FLOOR WEIJMA  
TITLE: GOOD JOB TEAM!

2021-06-09 21:28  
SENDER: DICK HEIJINK  
TITLE: SUBMISSION RECEIVED

2021-06-09 21:28

SUBJECTS

Q1 | Q2 | Q3 | Q4

INTRODUCTION TO PROGRAMMING  
IT AND LAW  
COMPIERS AND OPERATING SYSTEM  
DEVELOPMENT TOOLS

EXAM SCHEDULE

2021-06-09 21:28 HELLO WORLD

SCHEDULE

2021-06-09 21:28 C++  
2021-06-09 21:28 CURIOSITY  
2021-06-09 21:28 CONCURRENCY  
2021-06-09 21:28 IT AND LAW

FIND

TEACHER NAME

NAME: DICK HEIJINK  
EMAIL: J.D.HEIJINK@SAXION.NL  
ABBREVIATION: JDH

NAME: FLOOR WEIJMAN  
EMAIL: F.R.WEIJMAN@SAXION.NL  
ABBREVIATION: FRW

NAME: ROGIER HOMMELSEN  
EMAIL: R.HOMMELSEN@SAXION.NL

GRADES

YEAR 1 | YEAR 2 | YEAR 3 | YEAR 4

DEV TOOLS 9

HELLO WORLD 10

WEB APPLICATIONS 8

C++ 10

Vaardigheden Paspoort  
Naga Baker 418136

Organisational Processes  
Team Work  
Conflict handling  
Infrastructure  
Communication  
English  
Dutch  
Presentation  
Hardware Interfacing  
Creative Technology

This spider web shows which competencies you have achieved.

According to the client requirement, we implemented spider map. All the data in map is based on calculations. There are a total of 9 courses here, and we have set up 3 types, each of type corresponds to several tags in the spider map. When the score of the course is more than 5(that means student pass the course), the system will determine the type of course and in the spider map, The value of the corresponding tags will plus 1.

For example, course "development tool". The "development tool" is a type 1 course, which means that if student pass this course(grade>5), students will get 1 point on "English", "Hardware interfacing", "creative Technology" and "infrastructure" in the spider map.

### peer study

The screenshot shows a web-based learning platform interface. At the top, there is a header bar with the 'SAXION' logo, navigation links for 'INTRO', 'ONLINE LECTURES', 'DISCUSSION', 'WEEK 1', 'WEEK 2', 'WEEK 3', and 'EXTRA', and a search bar indicating the URL 'localhost:8080/course'. Below the header, there is a sidebar on the left with icons for 'INSTRUCTION', 'ASSIGNMENT', 'PEER STUDY' (which is highlighted in green), 'VIDEO', 'SUBMIT', and 'QUIZ'. The main content area displays four assignment cards for 'WEEK 1': 1. 'Finish Hello world assignment 1' (Check the work of your classmate, VIEW button). 2. 'Finish Hello world assignment 2' (Check the work of your classmate, VIEW button). 3. 'Finish Hello world assignment 3' (Check the work of your classmate, VIEW button). 4. 'Finish Hello world assignment 4' (Check the work of your classmate, VIEW button).

According to customer requirements, peer study, quiz and progress bar in introduction page are interrelated. Students must complete all assignments in order and upload their work to the peer study.



The screenshot shows the SAXION Learning Management System (LMS) interface. At the top, there is a navigation bar with links: INTRO, ONLINE LECTURES, DISCUSSION, WEEK 1, WEEK 2, WEEK 3, and EXTRA. Below the navigation bar, there is a sidebar on the left with icons for INSTRUCTION, ASSIGNMENT, PEER STUDY, VIDEO, SUBMIT, and QUIZ. The main content area displays two assignment sections: "Finish Hello world assignment 1" and "Finish Hello world assignment 2". Each section has a "VIEW" link, a "Check the work of your classmate" button, and a "Upload your work" section. In the "Upload your work" section for assignment 1, there is a file input field containing "filename (6).txt (231 B)", a "UPLOAD" button, and a "Choose a classmate to give you feedback" section with two options: "Yang" and "Mykhailo". Red arrows point from the text annotations to specific elements: one arrow points to the "filename (6).txt (231 B)" text with the label "choose the assignment you finished"; another arrow points to the "UPLOAD" button with the label "upload"; and a third arrow points to the "Yang" and "Mykhailo" options with the label "choose who will have permission to provide feedback".

During the upload process, they can choose who will have permission to provide feedback on their work.

The screenshot shows the SAXION Learning Management System (LMS) interface. At the top, there is a green navigation bar with tabs: INTRO, ONLINE LECTURES, DISCUSSION, WEEK 1, WEEK 2, WEEK 3, and EXTRA. On the left, a sidebar menu includes: INSTRUCTION, ASSIGNMENT, PEER STUDY (which is currently selected and highlighted in green), VIDEO, SUBMIT, and QUIZ. The main content area displays a "Finish Hello world assignment 1" section. It says "Check the work of your classmate". Below this, under "VIEW", there is a message: "Assignment was successfully uploaded" with a green checkmark. A red arrow labeled "1." points to this message, with the text "you already upload your work" written in red next to it. Below this, a student profile "Yang" is shown with a "DOWNLOAD" button. A red arrow labeled "2." points to the "Comments: 2" section, which lists two comments from "Mykhailo" and "Yang". The "Mykhailo" comment says "Good job!" and the "Yang" comment says "Well done!". Below the comments, there is a text input field with a pen icon and the placeholder "Enter your comment here", followed by a "SEND" button. A red arrow labeled "3." points to this input field. At the bottom of the page, there is a file download link: "filename (7).txt" with a download icon, and a red arrow points to it with the text "download other classmate work".

When they submit their homework, the system will display the work of other students, they can download, view, and then write comments.

The screenshot shows a digital assignment submission interface. At the top, there is a green header bar with the word "SAXION". Below it, a text input field contains "dssfdsf". Underneath, there is a comment section with a text input placeholder "Enter your comment here" and a green "SEND" button. A red arrow labeled "1." points to the "SEND" button, and the text "send comment" is written next to the name "Mykhailo". Below this, there is a "DOWNLOAD" button. The next section shows a comment from "Mykhailo" with a "DOWNLOAD" button. A red arrow labeled "2." points to the "VIEW" button under the assignment "Finish Hello world assignment 2", with the text "assignment2 become available." written next to it. This section also includes a "VIEW" button. The final section shows a comment from "Mykhailo" with a "VIEW" button.

After the student give other classmate comment or feedback on assignment1, assignment2 will become available. Only after completing all assignments in the peer study, quiz will be unlocked, and students can start quiz.

#### *Quiz & Progress bar*

When students complete the peer study, they can start quiz. Quiz is the last task of each week, so when a student pass quiz, it means that he completes the week of study.

INTRO ONLINE LECTURES DISCUSSION WEEK 1 WEEK 2 WEEK 3 EXTRA

**Week 1 quiz**  
level: easy

0%

Step 1 Step 2 Step 3 Step 4 Step 5

Which of the following cannot be used as a loop condition.

A: i++; B: void aMethod(); C: bEqual = str.equals("q"); D: count = =;

**CONTINUE**

quiz supports 4 question types, single choice, multiple choice, short answer question and true or false question. Because client mentioned that the teacher is not responsible for checking everyone's quiz, so when after the student completes quiz, the score will be automatically calculated and displayed.

We set the passing grade to 3 and connect it to the progress bar in the introduction, if student pass this quiz( $grade > 3$ ), the progress bar will automatically move to the next week, if they fail ( $grade \leq 3$ ), they can retake quiz.

## 1. Fail quiz

INTRO ONLINE LECTURES DISCUSSION WEEK 1 WEEK 2 WEEK 3 EXTRA

**Week 1 quiz**  
level: easy

100%

Step 1 Step 2 Step 3 Step 4 Step 5

**X** Insufficient! Your score is: 1 **RETAKE**

student score

click button to retake quiz

## 2. Pass quiz

INTRO ONLINE LECTURES DISCUSSION WEEK 1 WEEK 2 WEEK 3 EXTRA

**Week 1 quiz**  
level: easy

back to intro

100%

Step 1 Step 2 Step 3 Step 4 Step 5

**✓** Congratulation! Your score is: 5

pass the quiz

The screenshot shows the Blackboard interface for the 'Introduction' module. At the top, there are tabs: INTRO, ONLINE LECTURES, DISCUSSION, WEEK 1, WEEK 2, WEEK 3, and EXTRA. Below these, under the 'DISCUSSION' tab, is a section titled 'Learning progress'. A horizontal progress bar has a green dot at the 'Week2' position. A red arrow points from the text 'if pass the quiz, progress bar will move to next week' to this green dot.

**Introduction**

if pass the quiz, progress bar will move to next week

Hello students (that participated in the retake of an Introduction to Programming)! In a few moments the final grades for Introduction Programming retake of January 12th 2021 will be published via Bison. For most of you the results will not be a surprise as the individual rubric scores have been on Blackboard for some time now. We would like to ask all students to check if we have entered your grade correctly based on the number of points. Blackboard is leading in this, so if your grade somehow is incorrect, please contact me (Tristan, t.pothoven@saxion.nl) as soon as possible. As far as the exam review is concerned, Introduction Programming will \*not\* make use of the scheduled exam review time in quartile 3. This is because your grade is already explained via Blackboard including our arguments. It is of course possible that you have questions about your result or that you suspect an error has been made somewhere. You can report this by sending me (Tristan) an email. I do ask however that you provide the following information: Which rubric line is it about? (functionality, data types, methods, if-statements, loops or lists) What is wrong with it? (unclear reasoning from us, too few points for reason X, etc.) If you have questions on more than one topic, I would kindly request that you write this out per topic. General remarks such as "I disagree with the result" without any further argumentation unfortunately cannot be discussed. So please be as explicit as possible. You will have 2 weeks to respond to your result in case of questions. This means that the official period "exam review" now has begun and lasts until February 17th, 2021 after which all results are final. If you have any questions about the test or your grade, please feel free to send me an e-mail!

## Discussion

There are 2 channels on the discussion page, one is called "homework". Students can share the difficulties encountered in homework and help each other in this channel.

The screenshot shows the Blackboard interface for the 'Homework' module. At the top, there are tabs: INTRO, ONLINE LECTURES, DISCUSSION, WEEK 1, WEEK 2, WEEK 3, and EXTRA. Below these, under the 'DISCUSSION' tab, is a section titled 'HOMEWORK'. A user named 'YC' has posted a message: 'Can someone teach me Jasmine?'. The message includes a timestamp '00:18:19 GMT+0200 [2021-01-12T08:18:19Z]'. Red annotations with arrows point to several elements: 'Enter comments here' (pointing to the message input field), 'type here' (pointing to the message input field), 'send message' (pointing to the 'SEND' button), and 'the people who send message' (pointing to the user icon 'YC').

**HOMEWORK**

YC

Enter comments here

type here

send message

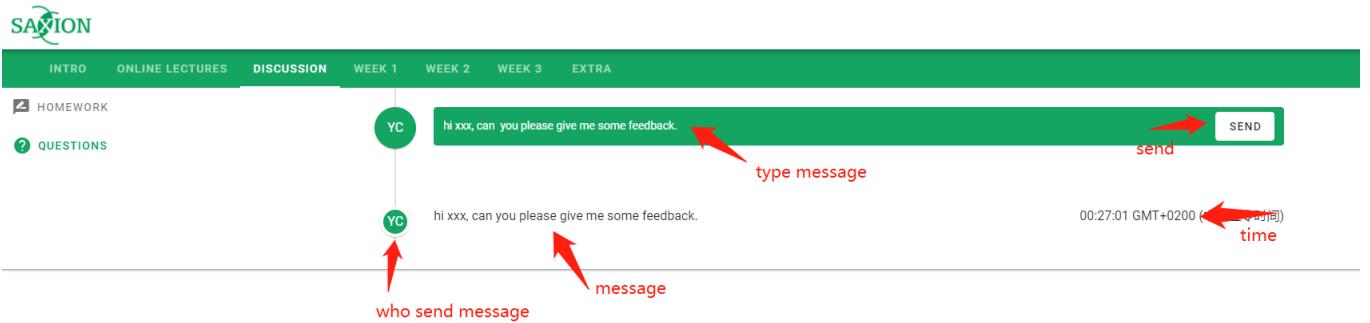
YC

Can someone teach me Jasmine?

00:18:19 GMT+0200 [2021-01-12T08:18:19Z]

the people who send message

Another is "questions", which is a platform for students and teachers to communicate. In this channel, teachers guide students to solve problems.



## Lectures

In the online lecture page, students can attend online classes, prepare for exams or review exams.

The screenshot shows the 'ONLINE LECTURES' page with a green header bar. Under the 'LECTURES' tab, there is a section for 'Week 1' with a 'JOIN' button. A red arrow labeled 'join the online class' points to this button. Below it is an 'Exam preparation' section with a 'JOIN' button. Under the 'ARCHIVE' tab, there is an 'Exam review' section with a 'JOIN' button. The entire page has a light gray background with horizontal lines separating sections.

they can also watch the recording of the course through the "archive".

INTRO    ONLINE LECTURES    DISCUSSION    WEEK 1    WEEK 2    WEEK 3    EXTRA

▶ LECTURES    **Week 1**  
📅 ARCHIVE    6/1/21, 11:45 AM - 6/1/21, 2:00 PM  
  
Duration: 1h 43m  
  
▶ WATCH  play recording

---

Week 2  
6/14/21, 11:45 AM - 6/14/21, 2:00 PM  
  
Duration: 46m  
  
▶ WATCH

## PROJECT PORTFOLIO

### Code of Conduct

1. All members should deliver their part.
2. All members should finish all tasks that are assigned to them.
3. All members should actively look for tasks when they are done with their current task.
4. All members should attend daily meetings on Teams.
5. Under the circumstances that a member cannot attend a meeting, (if possible) they don't need to inform the team in advance.

### Scrum-artefacts

#### Sprint 1

##### *Scrum retrospective*

Overall, the whole team worked well with each other as we attended all meetings and also discussed problems on discord outside class. Every teammates did their best and fulfilled their responsibilities. Although Keith and Tuan did not have much experience with Vuejs, they made a demo app following a tutorial on Youtube and caught up with the team. Whilst Yang and Mykhailo already have in-depth knowledge about Vuejs, they got to work on their parts and finished some small features. Sefanja did a great job with the wireframes while Jane did her part in completing the PoA as well as coding the base project. For next sprint, we will have to try to mind the deadlines more because sometimes the progress was still behind.

*Sprint's backlog*

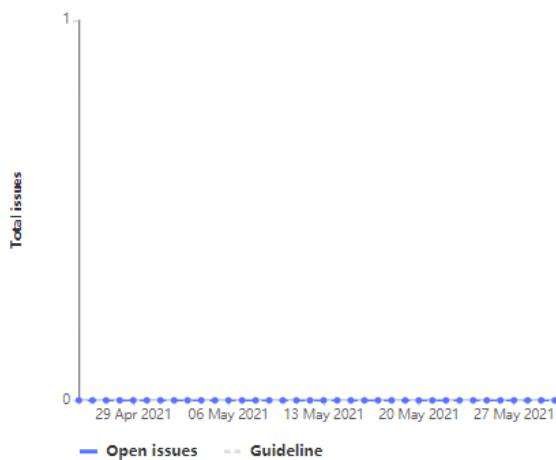
There is no sprint backlog for this sprint.

## Burndown chart

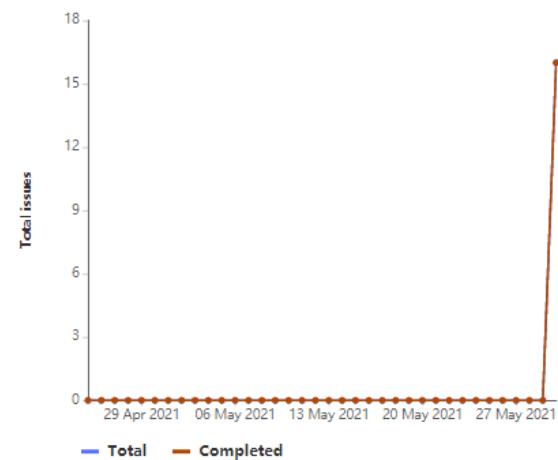
### sprint 1

Filter by [Issues](#) [Issue weight](#)

Burndown chart



Burnup chart



All issues for this milestone are closed. You may close this milestone now.

Issues 16 Merge requests 0 Participants 6 Labels 0

## Scrum related reports

Timesheet file is included.

2.5 IT Corp									
weekly hours goal 7-12 hours per week									
Total hours		wk1	wk2	wk3	wk4	wk5	wk6	wk7	wk8
yang	60.1	12.25	0	16.25	21.8	9.8	0	0	0
jane	23.6	10.75	0	9.25	1.8	1.8	0	0	0
tuan	73.6	13.25	0	14.75	31.8	13.8	0	0	0
sefanja	54.6	10.25	0	15.25	14.3	14.8	0	0	0
mykhailo g	66.1	16.25	0	20.25	14.8	14.8	0	0	0
keith	60.1	11.75	0	14.75	16.8	16.8	0	0	0
Total	338.1	74.5	0.0	90.5	101.3	71.8	0.0	0.0	0.0
PLEASE DO NOT WRITE YOUR HOURS ON THIS PAGE. YOU NEED TO WRITE IT IN TIMESHEET TAB. IT WILL CALCULATE IT HERE									
daily meeting attendance									
Total attendance		wk1	wk2	wk3	wk4	wk5	wk6	wk7	wk8
yang	23	5	0	10	4	4	0	0	0
jane	23	5	0	10	4	4	0	0	0
tuan	23	5	0	10	4	4	0	0	0
sefanja	23	5	0	10	4	4	0	0	0
mykhailo g	23	5	0	10	4	4	0	0	0
keith	23	5	0	10	4	4	0	0	0
*week 3 includes 2 weeks because vakantie week									

Standup Notes file is included.

## Sprint 2

### *Scrum retrospective*

This sprint, we create a total of 39 issues. So far, we have completed 37 issues and the remaining issue#55 and issue#52 will be placed in the next sprint.

### *What going well:*

In this sprint, we basically completed a basic website according to the requirements of client.

Although some areas still need to be improved, we continue to complete various components according to the client feedback and use Vuetify to make them more beautiful, to satisfy client. We also communicated with first year student to get their requirements and feedback on our web pages. Not only the front-end, we also completed the construction of our back end and database. The whole project is progressing smoothly, and we have also completed most of the plans we made earlier for this sprint.

### *What not going well:*

Our team had disputes and disagreements during this sprint. Unfortunately, because we did not handle the dispute well, the members of the team finally left, so now the team is divided. Myhailo and I decided to stay in the original team, while the others chose to leave and make a new team. The premise of group cooperation is that each group member respects each other, rather than letting all group members listen to him.

### *What about next sprint:*

*In the next sprint, Mykhailo and I will do our best to complete the entire project.*

### *Sprint's backlog*

Our team was able to finish all issues except for 2.

One of the unfinished issues is 50% done.

User Story 1	Priority: 1	MUST
Title	News component in home page	

Description	As a student, I want to know news about the school and have a button that allows me to view the details of the news.
Definition of done	We completed the news component, move the mouse to the icon, it will show how many news there are. Click the title, the latest news will be displayed.

User Story 2	Priority: 1	MUST
Title	Subjects component in home page	
Description	As a student, I want to be able to see all of my enrolled courses. Clicking on the course will jump to the course page.	
Definition of done	We have implemented the subject component and the course page, clicking on the course will jump to the course page.	

User Story 3	Priority: 1	MUST
Title	Exam schedule component in home page	
Description	As a student, I want to be able to clearly see the schedule of all the exams I need to take.	
Definition of done	Exam schedule can show how many exams there are. When you click on the title, a small calendar will appear. The current date will be marked in green, and the test date will be marked in red.	

User Story4	Priority: 1	MUST
Title	Schedule component in home page	
Description	As a student, I want to see all my course schedules so that I can better arrange my study and free time.	
Definition of done	There is a schedule component on the homepage, but we have not implemented it yet.	

User Story 5	Priority: 1	MUST
Title	Email component in home page	
Description	As a student, I want to have a component that allows me to read emails, so that I can quickly read the teachers or classmates email.	
Definition of done	There is an email component on the homepage, but we have not implemented it yet.	

User Story 6	Priority: 1	MUST
Title	Find component in home page	
Description	As a student, I want to be able to see the email addresses of all teachers so that I can quickly contact my teachers.	
Definition of done	There is a find component on the homepage, but we have not implemented it yet.	

User Story 7	Priority: 1	MUST
Title	Grades component in home page	
Description	As a student, I hope to be able to clearly see the results of all my subjects and show my learning status through the spider map.	
Definition of done	we implemented spider map. All the data in map is based on calculations.	

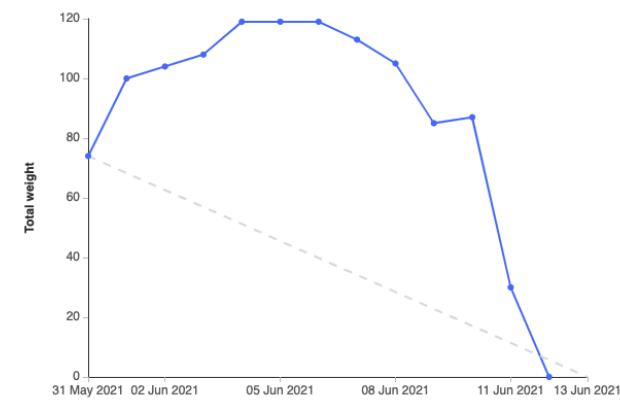
### Burndown chart

We tried to work with git as intended, using boards, issues, merge requests and more in sprint 2. All issues have been minimized for better workload separation and so we don't have weight 20 issues. The result you can see in the charts below.

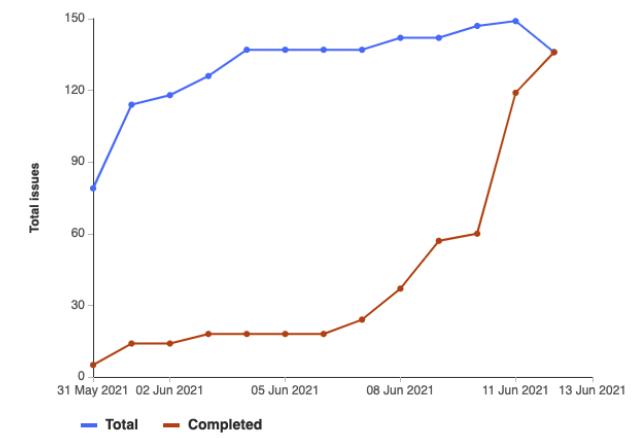
### sprint 2

Filter by Issues Issue weight

Burndown chart



Burnup chart



### Scrum related reports

## Timesheet overview

M16	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1					2.5 IT Corp									
2					weekly hours goal 7-12 hours per week								0.00	
3		Total hours		wk1	wk	wk2	wk3	wk4	wk5	wk6	wk7	wk8		
4	yang	118.7	vc	12.25	0	16.25	21.8	11.8	22.8	33.8	0	0		
5	jane	27.2		10.75	0	9.25	1.8	1.8	1.8	1.8	0	0		
6	tuan	77.2		13.25	0	14.75	31.8	13.8	1.8	1.8	0	0		
7	sefanja	58.2		10.25	0	15.25	14.3	14.8	1.8	1.8	0	0		
8	mykhailo g	116.7		16.25	0	20.25	14.8	14.8	19.8	30.8	0	0		
9	keith	70.8		11.75	0	14.75	16.8	16.8	9.8	0.9	0	0		
10														
11														
12	Total	468.8		74.5	0.0	90.5	101.3	73.8	57.8	70.9	0.0	0.0		
13														
14	PLEASE DO NOT WRITE YOUR HOURS ON THIS PAGE. YOU NEED TO WRITE IT IN TIMESHEET TAB. IT WILL CALCULATE IT HERE													
15														
16				daily meeting attendance										
17		Total attendance		wk1	wk	wk2	wk3	wk4	wk5	wk6	wk7	wk8		
18	yang	31		5	0	10	4	4	4	4	0	0		
19	jane	31		5	0	10	4	4	4	4	0	0		
20	tuan	31		5	0	10	4	4	4	4	0	0		
21	sefanja	31		5	0	10	4	4	4	4	0	0		
22	mykhailo g	31		5	0	10	4	4	4	4	0	0		
23	keith	29		5	0	10	4	4	4	2	0	0		
24														
25														
26	*week 3 includes 2 weeks because vakantie week													
27														
~														

Timesheet file is included.

Standup Notes file is included.

## Plan for coming sprint

The next sprint is the final sprint. Our goal is to first implement all the essential features that the customer wants. The following matters have the highest priority:

- Complete the content and functionality on the module page
- Link spider web to the grades component
- Create a progress bar on the module page
- Create separate APIs for the module page and like them together
- Refactor the code
- Make sure the customer is satisfied!

Sprint 3

### *Scrum retrospective*

In this sprint, we set up a total of 52 issues. At the end of the project, we completed all issues and successfully completed all the requirements of client.

What going well:

Because there are only 2 people, each person has lots of work. Even if the pressure of two of us is great, we spend a lot of time on this project every day, including weekends. In every client meeting, we can show client lots of new or improved things and can finish all client's requirements quickly. At the end of sprint3, two of us finally completed the entire project. While we are working hard to complete new requirements, we are also constantly updating or optimizing what has been completed. For example, after we heard about the vuex store, we read the documentation, applied it to the project, and updated the previously completed code to keep the entire project in a maintainable state. We also test the completed function to ensure its quality.

Things that could have gone better:

- More communication with client.
- Because there are only two of us, everyone needs to complete a lot of issues. We also work on this project on weekends, so we can increase the number of meetings to avoid some merger problems. As a result, we sometimes lost the boundaries between personal life and the school project. So, we would expect a response from the other member in no time.

Our cooperation is more freedom, and we can arrange meetings temporarily.

### *Sprint's backlog*

User Story 1	Priority: 1	MUST
Title	Course page navigation	
Description	As a user, I want to use navigation on the course page, and the navigation will display the information I want.	
Definition of done	We merge our old page, both the top and left navigation of the new page are working.	

User Story 2	Priority: 1	MUST
Title	Spider map	
Description	As a student, I want to be able to understand my learning situation through the spider map. And can clearly show the strengths and weaknesses.	
Definition of done	Make spider map into Pop-up panel.	

User Story 3	Priority: 1	MUST
Title	Video function	
Description	<p>As a teacher, I want to put my recorded instructional video into the course page.</p> <p>As a student, I want teachers to provide more video demos.</p>	
Definition of done	We have added a video interface to the week page, which supports teachers to upload videos.	

User Story 4	Priority: 1	MUST
Title	Submit	
Description	As a student, I hope there is a place in the course page for me to deliver the final assignment.	
Definition of done	We added a submit interface to the week page.	

User Story 5	Priority: 1	MUST
Title	Quiz function	
Description	As a student, I hope to have a quiz(self-test) every week so that I can better understand my learning situation. The quiz can be retake and scored automatically.	
Definition of done	We have implemented the function of quiz, which can give a score immediately after the student completes it. If the score is less than 3 points, there will be a retake.	

User Story 6	Priority: 1	MUST
Title	progress bar on the module page	
Description	As a student, I hope to see the progress of my current course as soon as I enter the course page, which helps me review and control time.	
Definition of done	We have added a progress bar to the intro interface, which can clearly reflect the student's learning progress, because it is connected to quiz. Only when the student passes quiz, the progress bar will move to the next week.	

User Story 7	Priority: 1	MUST
Title	Online course	

Description	As a student, I hope that there is a place in the course page that allows me to enter the online course, and it should support reviewing course recording.
Definition of done	use vuetyfy to make an online course component.

User Story 8	Priority: 1	MUST
Title	Intro page progress bar	
Description	As a student, I hope to see the progress of my current course as soon as I enter the course page, which helps me review and control time.	
Definition of done	We have added a progress bar to the intro interface, which can clearly reflect the student's learning progress, because it is connected to quiz. Only when the student passes quiz, the progress bar will move to the next week.	

User Story 9	Priority: 3	COULD
Title	Login& register	
Description	As a student, I want a school website account, and can log in or register.	
Definition of done	Realize register and login function.	

User Story 10	Priority: 1	MUST
Title	Peer study	
Description	As a student, I hope to have a platform where I can share code or homework ideas with other students. Students can learn from each other.	
Definition of done	We have completed the peer study, students can download other people's works after uploading their own works, and they can also comment under the works.	

User Story 11	Priority: 1	SHOULD
Title	Peer study download function.	
Description	As a student, I hope that in the peer study, I can download other people's works.	
Definition of done	We have completed the download function and support various file types.	

User Story 12	Priority: 1	MUST
Title	Discussion page	
Description	As a student, I hope to have a built-in discussion platform, including two channels for discussing homework with classmates and discussing code issues with teachers.	
Definition of done	We have implemented the discussion function. Includes two channels, students or teachers can type, each message includes the sender, content, and time.	

User Story 13	Priority: 2	MUST
Title	Footer link	
Description	As a student, I would like to have links to other websites of the school, such as the library.	
Definition of done	make buttons in the footer clickable.	

User Story 14	Priority: 1	MUST
Title	Connection between components	
Description	As a student, I want to make learning orderly. I want the components in the course to be related to each other and set up in order. maybe lock some components at the beginning.	
Definition of done	We connect the peer study to quiz. The correct order is that after completing the peer study, quiz will be unlocked, and quiz connect to the progress bar in the intro page.	

User Story 15	Priority: 1	MUST
Title	Assignment page	
Description	As a student, I hope to have a clear assignment page, preferably with a picture and a rating bar indicating the difficulty, so that I can better arrange my homework plan.	
Definition of done	We completed the assignment page and used the rating bar to indicate the difficulty of each assignment.	

User Story 16	Priority: 1	MUST
---------------	-------------	------

Title	System failure error message
Description	As a user, I hope to get the system's error description in time when the server fails.
Definition of done	We have completed the error message report. When the front end does not get the data from the back end after 5 seconds, an error information will be displayed.

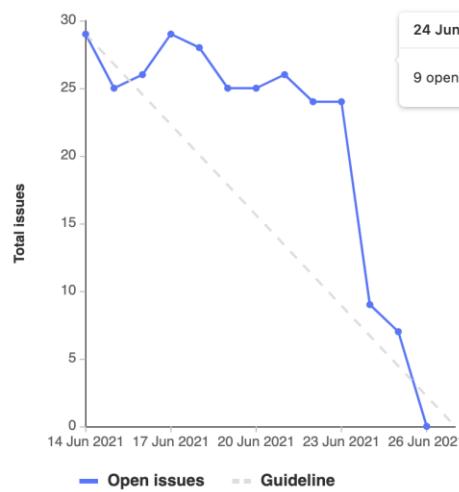
#### Burndown chart

Like in the previous sprint, we worked with GitLab as much as possible. In this sprint, we had 52 issues. For most issues we had a merge request. This approach helped us better maintain our code.

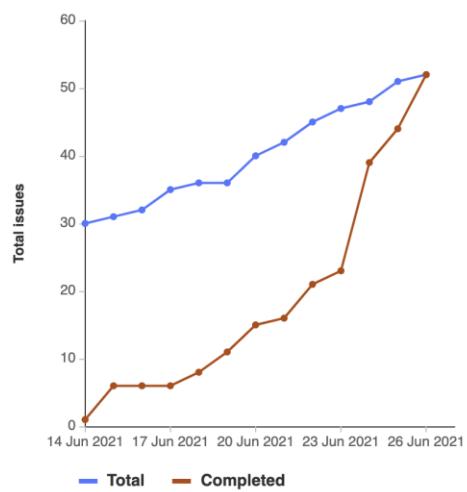
### sprint 3

Filter by Issues Issue weight

Burndown chart



Burnup chart



## Scrum related reports

### Timesheet overview

	A	B	C	D	E	F	G	H	I	J	K	L	M	N			
1					2.5 IT Corp												
2					weekly hours goal 7-12 hours per week												
3		Total hours			wk1	vk	wk2	wk3	wk4	wk5	wk6	wk7	wk8				
4	yang	181.1			12.25	0	16.25	21.8	11.8	22.8	33.8	35.15	27.3	-3.00			
5	jane	27.2			10.75	0	9.25	1.8	1.8	1.8	0	0					
6	tuan	77.2			13.25	0	14.75	31.8	13.8	1.8	1.8	0	0				
7	sefanja	58.2			10.25	0	15.25	14.3	14.8	1.8	1.8	0	0				
8	mykhailo g	170.6			16.25	0	20.25	14.8	14.8	19.8	30.8	24.15	29.8				
9	keith	70.8			11.75	0	14.75	16.8	16.8	9.8	0.9	0	0				
10																	
11																	
12	Total	585.1			74.5	0.0	90.5	101.3	73.8	57.8	70.9	59.3	57.0				
13																	
14	PLEASE DO NOT WRITE YOUR HOURS ON THIS PAGE. YOU NEED TO WRITE IT IN TIMESHEET TAB. IT WILL CALCULATE IT HERE																
15																	
16					daily meeting attendance												
17		Total attendance			wk1	vk	wk2	wk3	wk4	wk5	wk6	wk7	wk8				
18	yang	45			5	0	10	4	4	4	4	7	7				
19	jane	31			5	0	10	4	4	4	4	0	0				
20	tuan	31			5	0	10	4	4	4	4	0	0				
21	sefanja	31			5	0	10	4	4	4	4	0	0				
22	mykhailo g	45			5	0	10	4	4	4	4	7	7				
23	keith	29			5	0	10	4	4	4	2	0	0				
24																	
25																	
26																	
27																	
	*week 3 includes 2 weeks because vakantie week																

Timesheet file is included.

Standup Notes file is included.

### Proof of Concept

We use SQLite database, get data through API, and send the data to the front end through fetch API.

//Manual instruction

Step 1: open & run our project.

After unzipping and opening with IntelliJ, the project names we used are "vue-app3-front" and "vue-app3-back". One is our front-end and the other is the back end. After npm i to them, it can run normally.

To start frontend run in the vue-app3-front folder: **npm run serve**

To start backend run in the vue-app3-back folder: **npm run start**

Step 2: view our result.

After running the project, you can use the browser to view our work. The index page of the webpage is the home page(<http://localhost:8080/>). All data in the homepage is transmitted from SQLite to the front end through API. The module page is located in (<http://localhost:8080/course>)

The screenshot shows the SAXION student portal with several modules visible:

- NEWS**: Shows "latest news" items like "SAXION INTRODUCTION DAYS 2021: SAXION CAMPUS ALMOST OPEN" and "MAKE UP YOUR MINOR FIND OUT MORE ABOUT OUR OFFER NOW".
- SUBJECTS**: A "course module page" section for Q1 | Q2 | Q3 | Q4, listing subjects such as "INTRODUCTION TO PROGRAMMING", "IT AND LAW", "COMPILERS AND OPERATING SYSTEM", and "DEVELOPMENT TOOLS".
- EXAM CALENDAR**: A "exam calendar" section showing exam times for various subjects: HELLO WORLD (21:28), HBU-ICT PROJECT (21:28), WEB DEVELOPMENT (21:28), DEV TOOLS (21:28), and CONCURRENCY (11:45AM), IT AND LAW (14:30PM), and another CONCURRENCY entry (17:10PM).
- EMAIL PREVIEW**: A list of recent emails from DICK HELINK, FLOOR WELJMA, and ROGER HOMMELIN.
- FIND**: A search function for teachers, showing results for DICK HELINK, FLOOR WELJMA, and ROGER HOMMELIN.
- GRADES**: A "spider map" showing grades for different subjects across four years (YEAR 1 | YEAR 2 | YEAR 3 | YEAR 4). The subjects listed are DEV TOOLS, HELLO WORLD, WEB APPLICATIONS, C++, COMPLEXITY AND ALGORITHMS, and CONCURRENCY.

## ICT Card: Survey

Target number of questions 7 questions. Keep it close ended i.e multiple choice questions.

Survey link:

<https://docs.google.com/forms/d/1kasEtMpxj8tp0HyVemiCodY7BiDkjBaCLkntGvzWHGs/edit>

- Which self-studying helping features/functions do you want Blackboard to have? - Tuan Nguyen
  - Self-made schedule
  - Exam/Class reminders
  - Daily quiz
  - Multiple choice sample tests before exams
  - Other:...
- What do you think about the overall distance education of Saxion? - Sefanja
  - Poor
  - Below Average
  - Average
  - Good
  - Excellent
- Do you enjoy learning online? - Sefanja
  - Yes
  - It's alright
  - No
- In terms of online lectures, what do you think can improve in this environment? (Multiple answers can be selected) - Sefanja
- what do you think can be improved in Saxion's learning environment?
  - Recordings
  - Interactivity
  - Quality of online lecture

- Chat
  - Online whiteboard
  - Webcam functions
  - Share files functions
  - Screenshare functions
- What is wrong with blackboard - Keith
  - Nothing is bad
  - Bad design
  - Hard to navigate
  - Everything is bad
- Do you prefer online learning or offline learning – Keith
  - Online
  - Offline
  - Either is fine
  - Why?
- During online lectures, do you prefer to see other people via their cam – Keith
  - Yes
  - No
- Are you satisfied with the current digital education environment (blackboard)? - Yang
  - Yes
  - No
  - Why yes/no?
- Which the following aspects of blackboard do you think needs to be improved? -Yang
  - Schedule
  - course module
  - exam module
  - message module (teacher's feedback, Course announcement)
- Is it necessary to add some reminder functions to the blackboard, such as deadline reminder, class reminder and exam reminder? - Yang
  - Yes
  - No

- Do you think it is a good idea to change the blackboard homepage to a page that supports personalized editing (you can change the position, size, and color of each module)? -Yang
  - Yes
  - No
- Have you ever experienced interruption while taking exams/quizzes online via Blackboard? *Jane*
  - Very often
  - Often
  - Not too often
  - Never
- Which of the following is the main reason for causing drawbacks of using Blackboard? *Jane*
  - Blackboard UI sucks, mobile version not fully supported
  - Teachers being unorganized and/or inconsistent with the use Blackboard
  - Trouble with connections, experience interruptions when using Blackboard Collaborate
  - Layout is confusing for locating the necessary modules, and too much unnecessary things on one page
- Which of the following features do you used the most on Blackboard, besides looking at class materials/syllabi? *Jane*
  - Watch previously recorded lectures
  - Check grades
  - Use discussion board
  - Check class roster
  - Communicate and collaborate with team members
  - Get feedback from teachers on submitted assignments
- Which of the following LMS have you ever heard of besides Blackboard? *Jane*
  - Canvas
  - Moodle
  - Edmodo

- Google Classroom
- None of the above

## Meetings

### Questions for client

- What kind of inspiration is available. What has been thought of already. Who is working on it
- Do we need acceptance testing, user testing, software testing
- Does it needs to have functionalities like blackboard assignment receiving or is it for attending class
- Are there any forbidden things? (example: Saxion doesn't want to be able to force students webcams on)

### Client Meetings

#### 26 Apr 2021

- Create survey

#### 28 Apr 2021

- Use research cards (<https://www.cmdmethods.nl/>) / Design thinking
- Presentation on 29 June for client / prototype
- Web Application of my schedule

#### 11 May 2021

- Client asked to check POA process
- POA form should be ready for meeting on Wednesday

#### 19 May 2021

- Client showed us Coursera website as an example how the online classes look like on that website to give us some ideas

#### 26 May 2021

- Client asked if we can show our project to first year students because they are not happy with the current system

#### 2 Jun 2021

- Client wants us to implement spider web, progress bar and come up with the content for the module page.

#### 9 Jun 2021

- More in-depth discussion for module content.

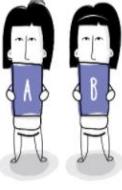
#### 16 Jun 2021

- Determine the mutual connection between the components and the order of completion, add a progress bar to the introduction page, and connect to quiz.

#### 23 Jun 2021



# ICT Research Methods

<p> <b>A/B Testing</b></p>  <p><b>Why?</b> A minor change in a design may alter user behaviour in ways that are hard to detect in a usability test. An A/B test allows you to compare real-world user behaviour across different versions of the product.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Concept</b></p>  <p><b>Why?</b> When you develop a new product or service, the concept summarizes 'the big idea' or 'the main principle' on which your solution will be based. For example, most traditional churches have a floor plan based on a cross so God can recognize a church from the sky. Validate your concept(s) with stakeholders to determine desirability and feasibility.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Customer Journey</b></p>  <p><b>Why?</b> Visualize the user experience of a service over time and across the different interaction moments (touch points) within the service.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Day in the life</b></p>  <p><b>Why?</b> Gain insights in the life of your users. To gain empathy, a day in the life could be a useful technique.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Ideation</b></p>  <p><b>Why?</b> Generate and develop new ideas.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>
<p> <b>Online analytics</b></p>  <p><b>Why?</b> Gain insights from real usage statistics in order to continue improving a website, app or social media campaign after it is in use, or monitor its use for marketing purposes.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Proof of Concept</b></p>  <p><b>Why?</b> Demonstrate the desirability or the feasibility of your idea or design.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Prototype</b></p>  <p><b>Why?</b> Test an early model of your product with users, peers, experts or your client. Test goals can vary from testing the concept, to testing functionality, user experience, content breakdown, usability, or technical feasibility.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Prototyping</b></p>  <p><b>Why?</b> Develop, evaluate or communicate a concept or design.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>	<p> <b>Scenario</b></p>  <p><b>Why?</b> Different types of scenarios exist that each serves a different purpose, for example to develop user requirements, to generate ideas or to reflect on a concept.</p> <p><a href="#">Remove</a> <a href="#">More info</a></p>

 Survey



**Why?**  
Collect, mostly quantitative, information from a large sample of your target group.

[Remove](#) [More info](#)

 Thinking aloud



**Why?**  
Understand the reasons behind user behaviour, or uncover the mental models of the user in a usability test.

[Remove](#) [More info](#)

 Usability Testing



**Why?**  
Detect problems users have with your design and correct these before the product goes live.

[Remove](#) [More info](#)

 USP (Unique Selling Points)



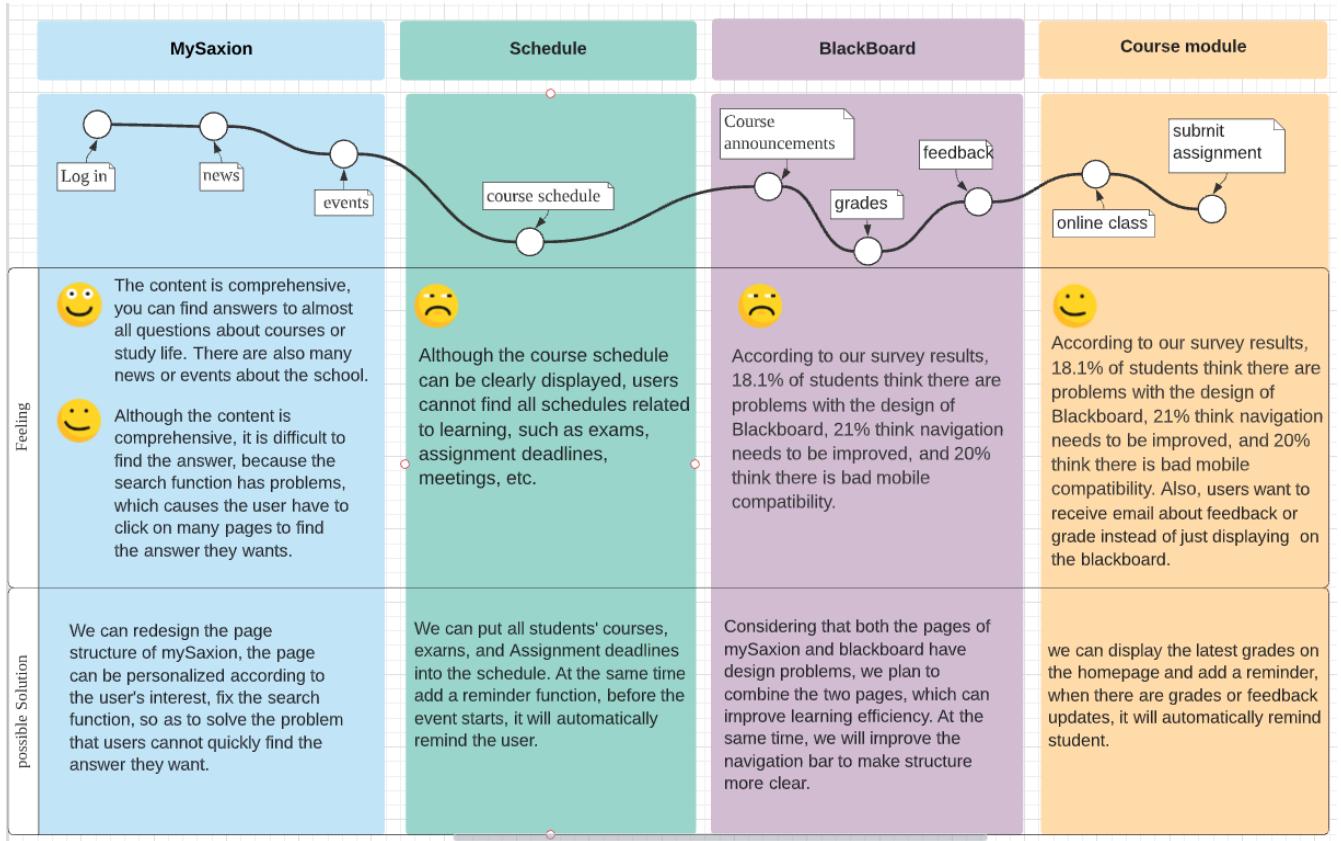
**Why?**  
In a competitive business situation you need to be able to identify and communicate clearly and concisely what it is that sets you apart.

[Remove](#) [More info](#)

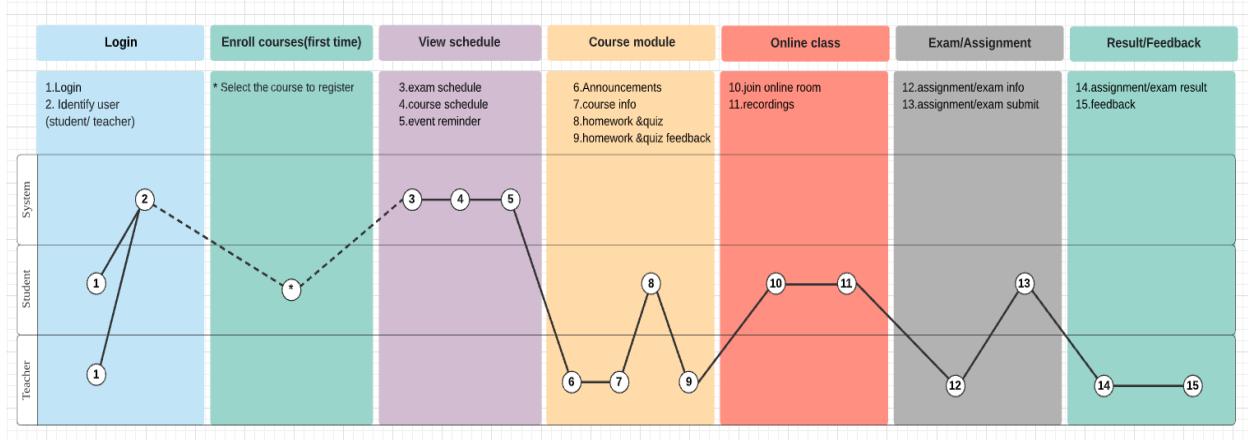
## ICT Card: Ideation

- Create a freecodecamp type of practice environment
- Create wireframe design for logged in students/teachers
- Something that applies to all of saxion students/department, better design
- Students should be able to pin “apps”(?) to their liking on MySaxion like start menu on Windows
- Components should be movable

## Customer Journey map

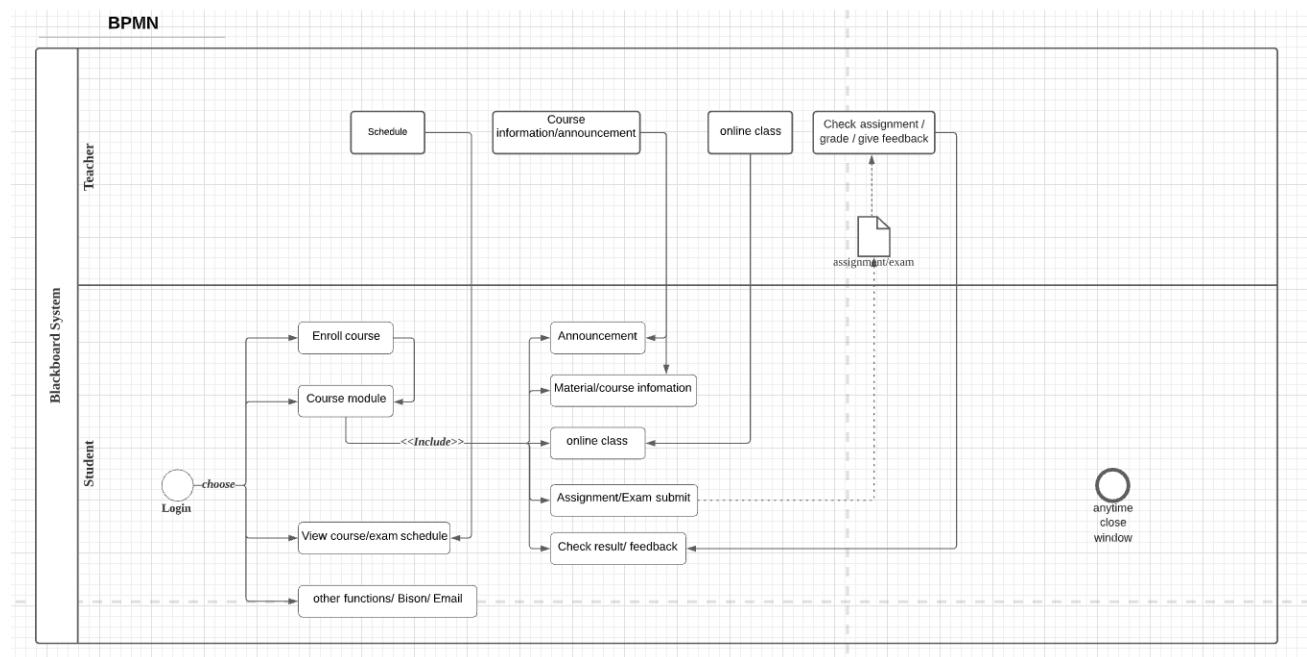


## Activity map



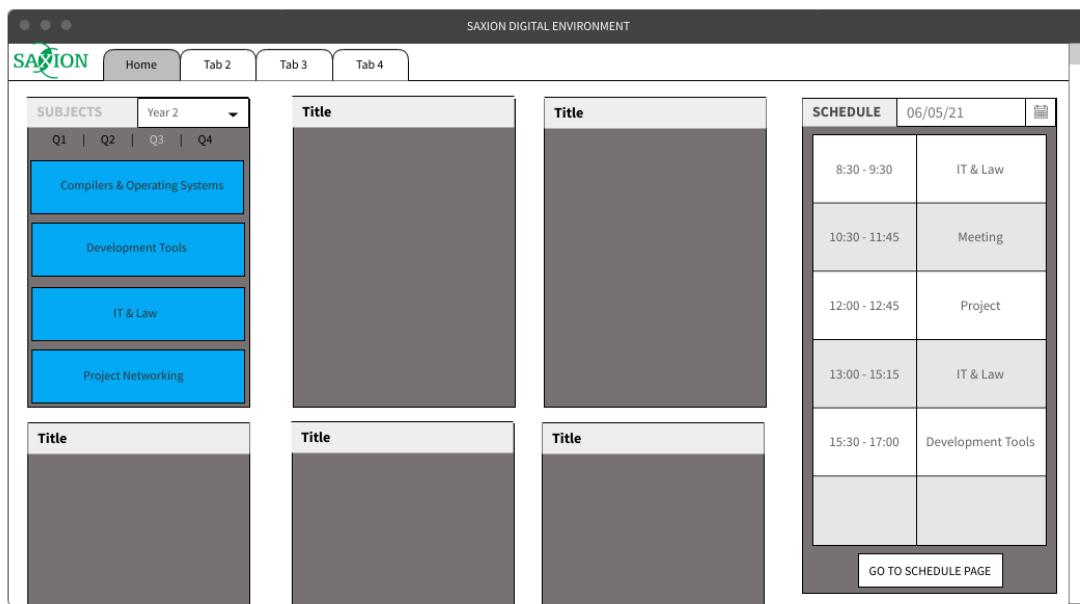
## BPMN

[https://lucid.app/lucidchart/3420cbcd-cc37-4797-bade-6a0875dce497/edit?page=0\\_0#](https://lucid.app/lucidchart/3420cbcd-cc37-4797-bade-6a0875dce497/edit?page=0_0#)



## Wireframes

1<sup>st</sup> Concept wireframe



Feedback:

- More titles

New ideas for titles:

- Exam schedule – Keith
- Class schedule – Yang
- Email preview - Keith
- Results window - Keith
- Search box – Yang
- Contact list box, tinder - Tuan
- News box – Keith
- Complain box – Keith
- Students chat box – Mykhailo
- Progress bar – Tuan
- Exam, class pop-up reminder - Tuan
- Student balie chat box – Keith
- Login & Register – Yang

## 2<sup>nd</sup> Concept wireframe

View/download full size wireframe from here: <https://imgur.com/a/4wDo1aZ>

### *Homepage*

**SAXION LEARNING ENVIRONMENT**

**SUBJECTS** YEAR 2

Q1 | Q2 | Q3 | Q4

**COMPILERS AND OPERATING SYSTEMS**

**DEVELOPMENT TOOLS**

**IT AND LAW**

**PROJECT NETWORKING**

**NEWS**

STUDENT FELL OF THE SAXION STAIRS AS STAIRS COLLAPSED. STUDENT IS IN HOSPITAL AND WILL BE OKAY. HOWE.. [READ ENTIRE ARTICLE](#)

AMAZING ACHIEVEMENT BY SAXION ICT STUDENTS. MADE NASA PROGRAM THESE AMAZING TALENTS WILL BE R.. [READ ENTIRE ARTICLE](#)

SAXION NOW OFFERS NEW SANDWICH IN CAFETERIA. STUDENT SEEM ADDICTED TO ... [READ ENTIRE ARTICLE](#)

[READ ALL NEWS](#)

**GRADES**

YEAR 1 | YEAR 2 | YEAR 3 | YEAR 4

2.1 CONCURRENCY	6
2.1 PROJECT	8
2.1 IT AND LAW	4
2.2 DEV TOOLS	2
2.2 WEB TECH	8
2.2 PROJECT	6
2.3 C++	5
2.3 PROJECT	3
2.4 PROJECT	9

**SCHEDULE** 12 May 2021

8:30 - 10:00	IT AND LAW
11:00 - 13:00	DEV TOOLS
13:30 - 14:45	PROJECT
16:00 - 17:00	MEETING

**FIND STUDENTS & TEACHERS**

TEACHER NAME  
2 RESULTS

X	TEACHER NAME EMAIL@SAXION.NL TEACHERCODE
X	TEACHER NAME EMAIL@SAXION.NL TEACHERCODE

**EXAM SCHEDULE**

24-06-21 13:00-12:00	IT & LAW
25-06-21 13:00-12:00	DEV TOOLS
30-06-21 13:00-12:00	CONCURRENCY
01-07-21 13:00-12:00	WEBTECH

**EMAILBOX PREVIEW**

SENDER: DATE: 12/06/21  
TITLE:

SENDER: DATE: 12/06/21  
TITLE:

[GO TO EMAILS](#)

[GO TO SCHEDULE PAGE](#)

[CHAT WITH STUDENTBALI](#)

*Course page*

The wireframe shows a web-based learning environment for 'INTRODUCTION TO PROGRAMMING'. The top navigation bar includes the SAXION logo, the title 'INTRODUCTION TO PROGRAMMING', and a 'BACK TO HOMEPAGE' button. Below the title, a horizontal menu bar lists 'INTRODUCTION', 'LECTURES', 'WEEK 1' (highlighted in green), 'WEEK 2', 'WEEK 3', 'WEEK 4', 'WEEK 5', 'WEEK 6', 'WEEK 7', 'WEEK 8', and 'EXTRA'. On the left, a sidebar contains links for 'INTRODUCTION', 'INSTRUCTION', 'VIDEOS' (highlighted in green), 'ASSIGNMENT', 'PEER TO PEER', and 'DISCUSSION BOARD'. The main content area features a large video player with a play button, a progress bar showing 3:06/1:46, and video controls. To the right of the video, a vertical list of video topics is displayed:

- 1. Introduction to variables
- 2. Int, String and char
- 3. Usage of variables
- 4. Typecasting
- 5. Summary of variables
- 6. Datatypes

3<sup>rd</sup> Concept wireframe

View/download full size image from here: <https://imgur.com/a/roWDmy2>

## Homepage

The screenshot shows the Saxon Learning Environment homepage with the following sections:

- SUBJECTS**: Includes links to COMPILERS AND OPERATING SYSTEMS, DEVELOPMENT TOOLS, IT AND LAW, and PROJECT NETWORKING.
- NEWS**: Headline: "STUDENT FELL OF THE SAXION STAIRS AS STAIRS COLLAPSED. STUDENT IS IN HOSPITAL AND WILL BE OKAY. HOWE..." with a "READ ENTIRE ARTICLE" button. Another news item: "AMAZING ACHIEVEMENT BY SAXION ICT STUDENTS. MADE NASA PROGRAM THESE AMAZING TALENTS WILL BE R..." with a "READ ENTIRE ARTICLE" button.
- GRADES**: Shows grades for YEAR 1, YEAR 2, YEAR 3, and YEAR 4. A red arrow points to the "2.1 CONCURRENCY" row in the YEAR 2 section, which has a value of 6.
- SCHEDULE**: Displays a weekly schedule from 8:30-10:00 to 16:00-17:00. The schedule includes IT AND LAW, DEV TOOLS, PROJECT, and MEETING.
- COMPETENCES**: A sidebar titled "GRADES" showing competence levels for COMPETENCE 1 through COMPETENCE 8, each represented by a star rating. A red arrow points to the "SEE COMPETENCES" button in the GRADES section.
- FIND STUDENTS & TEACHERS**: A search interface showing results for TEACHER NAME, EMAIL@SAXION.NL, and TEACHERCODE.
- EXAM SCHEDULE**: Lists exams for IT & LAW, DEV TOOLS, CONCURRENCY, and WEBTECH.
- EMAILBOX PREVIEW**: Shows two email entries with SENDER, TITLE, and DATE.
- COLOR SCHEME**: Displays color hex codes: #15a563, #ea1919, and #212121.

## Course page

The screenshot shows the Saxon Learning Environment course page for "INTRODUCTION TO PROGRAMMING".

**Navigation:** INTRODUCTION, LECTURES, WEEK 1, WEEK 2 (highlighted), WEEK 3, WEEK 4, WEEK 5, WEEK 6, WEEK 7, WEEK 8, DISCUSSION BOARD, EXTRA, BACK TO HOMEPAGE.

**Left Sidebar:** INTRODUCTION, INSTRUCTION, VIDEOS (highlighted), ASSIGNMENT, PEER TO PEER.

**Main Content:** A large video player area showing a video player interface with a play button, volume control, and a progress bar at 3:06/1:46:31. Below the video are six video thumbnails:

- 1. Introduction to variables
- 2. Int, String and char
- 3. Usage of variables
- 4. Typecasting
- 5. Summary of variables
- 6. Datatypes

**Bottom Right:** Buttons for CC, HD, settings, and full screen.

## Assignment page

The screenshot shows a web-based learning environment for 'INTRODUCTION TO PROGRAMMING'. The top navigation bar includes the SAXION logo, the title 'INTRODUCTION TO PROGRAMMING', and a 'BACK TO HOMEPAGE' button. Below the title, a horizontal menu bar lists 'INTRODUCTION', 'LECTURES', 'WEEK 1', 'WEEK 2' (which is highlighted in green), 'WEEK 3', 'WEEK 4', 'WEEK 5', 'WEEK 6', 'WEEK 7', 'WEEK 8', 'DISCUSSION BOARD', and 'EXTRA'. On the left, a vertical sidebar contains links for 'INTRODUCTION', 'INSTRUCTION', 'VIDEOS', 'ASSIGNMENT' (which is also highlighted in green), and 'PEER TO PEER'. The main content area is divided into two sections: 'ASSIGNMENT 1' and 'ASSIGNMENT 2'.  
**ASSIGNMENT 1:**  
Write a Java program to print the sum of two numbers.  
Test Data:  
74 + 36  
Expected Output :  
110  
**ASSIGNMENT 2:**  
Write a Java program to print the result of the following operations.  
Test Data:  
a. -5 + 8 \* 6  
b. (55+9) % 9  
c. 20 + -3\*5 / 8  
Expected Output :  
43  
1  
19

# Final concept wireframe

## Homepage

The wireframe displays the following sections:

- SUBJECTS**: Includes links for Q1 | Q2 | Q3 | Q4, COMPILERS AND OPERATING SYSTEMS, DEVELOPMENT TOOLS, IT AND LAW, and PROJECT NETWORKING.
- NEWS**: Headlines include "STUDENT FELL OF THE SAXION STAIRS AS STAIRS COLLAPSED. STUDENT IS IN HOSPITAL AND WILL BE OKAY. HOWE...", "AMAZING ACHIEVEMENT BY SAXION ICT STUDENTS. MADE NASA PROGRAM", and "SAXION NOW OFFERS NEW SANDWICH IN CAFETERIA. STUDENT SEEM ADDICTED". Buttons for "READ ARTICLE" and "SEE ALL NEWS" are present.
- GRADES**: A table with columns for SUBJECT and GRADE, showing 10 rows of data. Buttons for "YEAR 1" through "YEAR 4" are at the top.
- SCHEDULE**: A list of events from 08:30 to 08:30, including MEETING, CLASS, CLASS, CLASS, MEETING, TASK, SAXION EVENT, SAXION EVENT, and SAXION EVENT. A "SEE ENTIRE SCHEDULE" button and a message icon are at the bottom right.
- FIND**: A search bar for TEACHERNAME with a magnifying glass icon. Below it are two sections for searching by NAME, EMAIL, and TEACHERCODE.
- EXAM SCHEDULE**: A table with rows for 21/06/21 08:30 (IT AND LAW), 24/06/21 10:30 (DEV TOOLS), 26/06/21 11:30 (CONCURRENCY), and 30/06/21 14:00 (WEB TECH).
- EMAIL PREVIEW**: A list of five email entries with dates from 21/06/21 08:30, showing SENDER and TITLE for each.

## Individual Reflections

Attached in the delivery document.

## Declaration of Competences

Attached in the delivery document

Yang

Competence Choice 1: Realization

Competence Choice 2: Manage + Control

Competence Choice 3: Design

Mykhailo

Competence Choice 1: Realisation

Competence Choice 2: Advise

Competence Choice 3: Manage + Control

## Resources

### References

<https://doe-meer-met-studiedata.nl/wat-betekent-dit-voor-mij/>

Levels of competences

<https://hbo-i.nl/domeinbeschrijving/>

Todo applications

<https://todomvc.com/>

Som Model English

<https://som-eng.digitaal-magazine.nl/eng-som-2-0/educational-philosophy>

Final Survey link:

<https://docs.google.com/forms/d/1kasEtMpxj8tp0HyVemiCodY7BiDkjBaCLkntGvzWHGs/edit>

GitLab Repository

<https://gitlab.com/saxion.nl/hbo-ict/2.4-project/student-learning/activity>