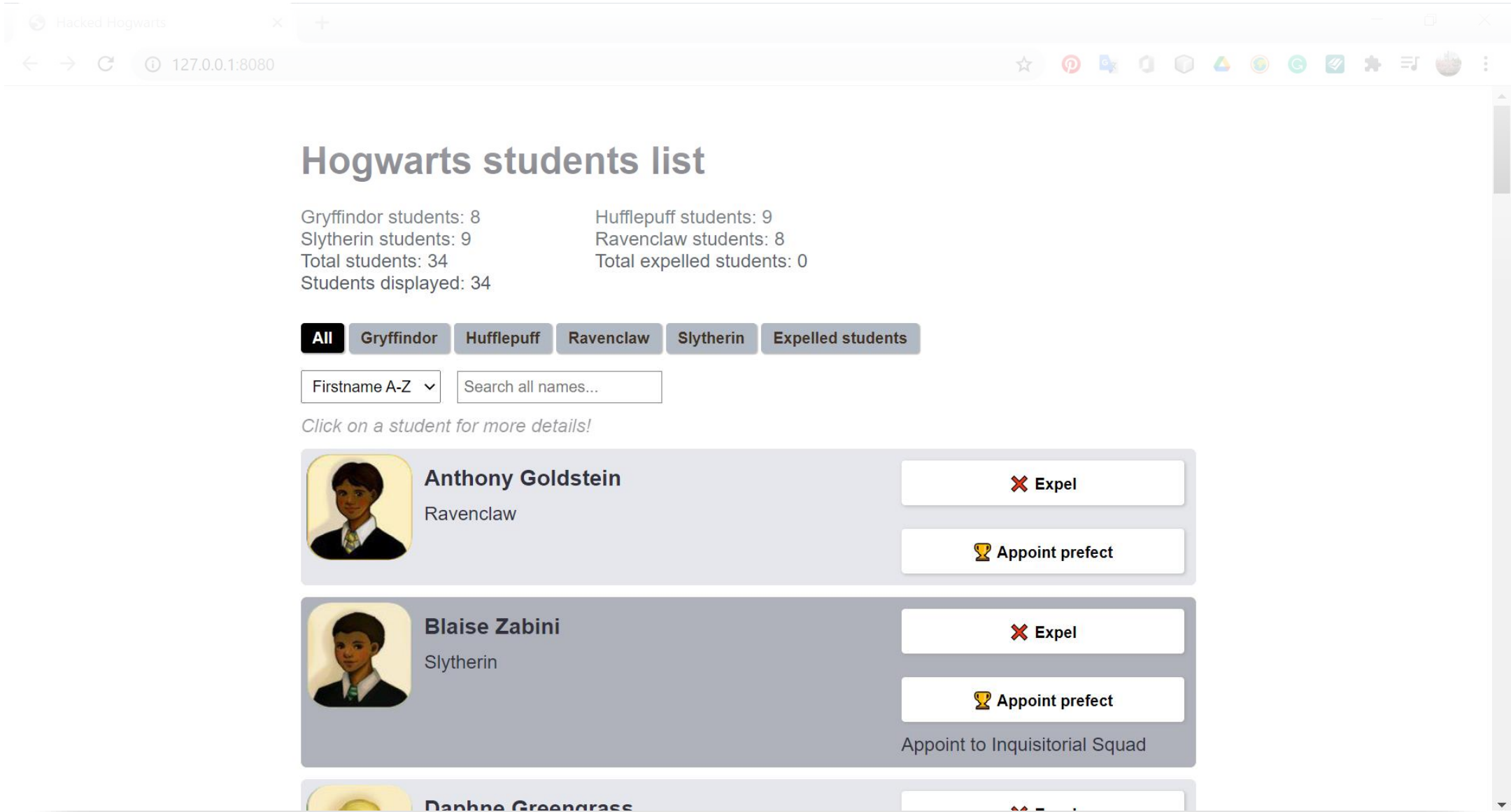


Hacked Hogwarts

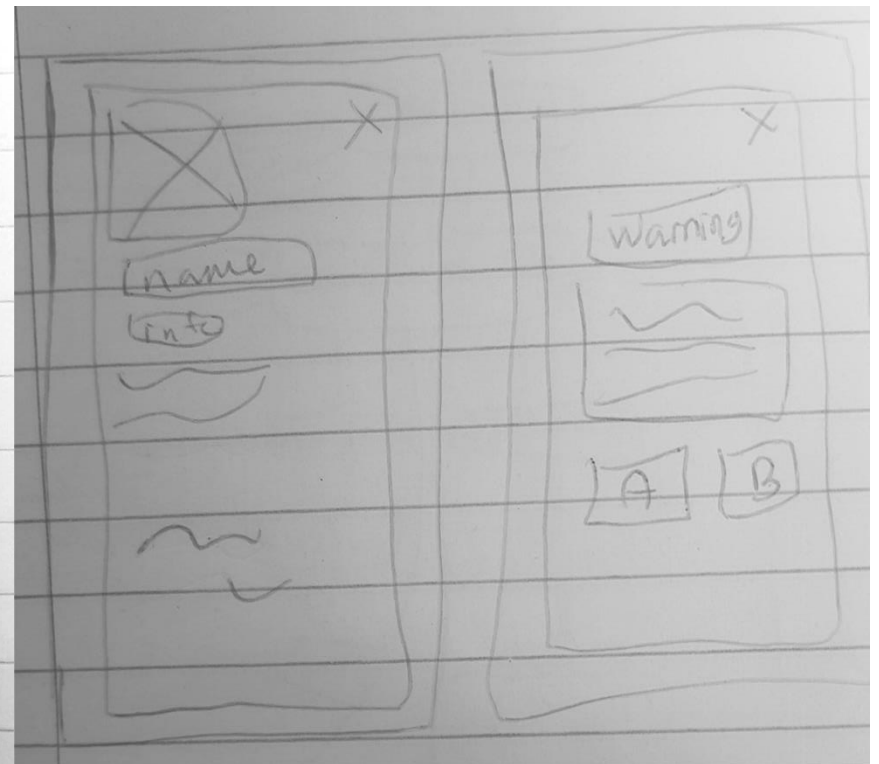
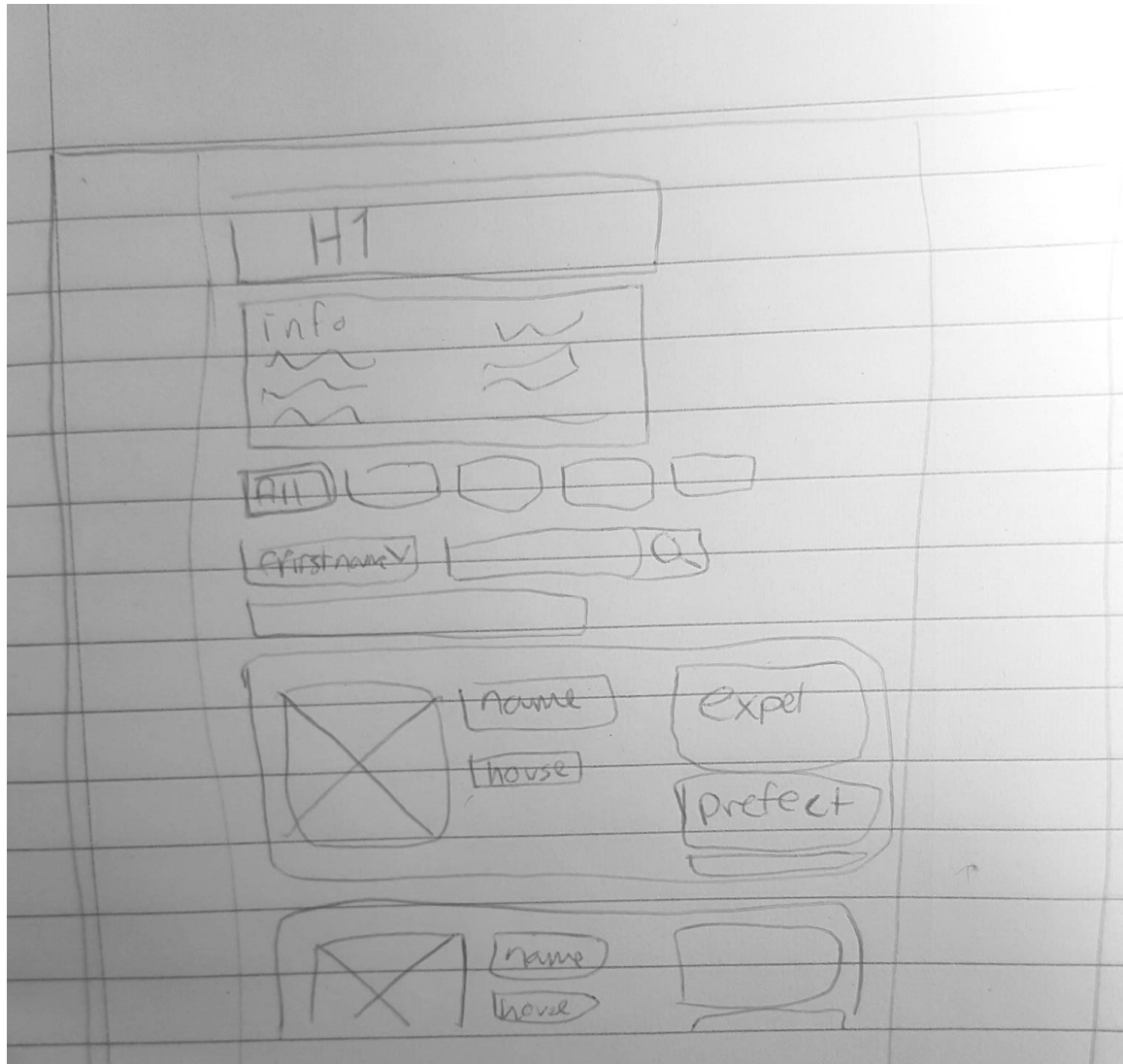
Sarah Isabella Frederiksen

Github repository: https://github.com/sara616b/hacked_hogwarts

Final solution: http://sarahfrederiksen.dk/kea/frontend/hacked_hogwarts/index.html



Initial wireframes for list view, popup, and warning modal:



Features/user stories

As an administrator, I want to be able to view a list of all the students in this year so that I can get a quick overview.

As an administrator, I want to be able to get a closer look at any student so that I can easily see everything about a student.

As an administrator, I want to be able to filter the list according to the students' houses so that I can get an overview over the status of each house individually without being distracted by all the other students.

As an administrator, I want to be able to sort the student list by first and last name so it's easy to locate a student, in line with I also want to be able to search for a student's name.

As an administrator, I want to be able to expel a student and have them removed from the overall list so that the list is always a reflection of the current student body.

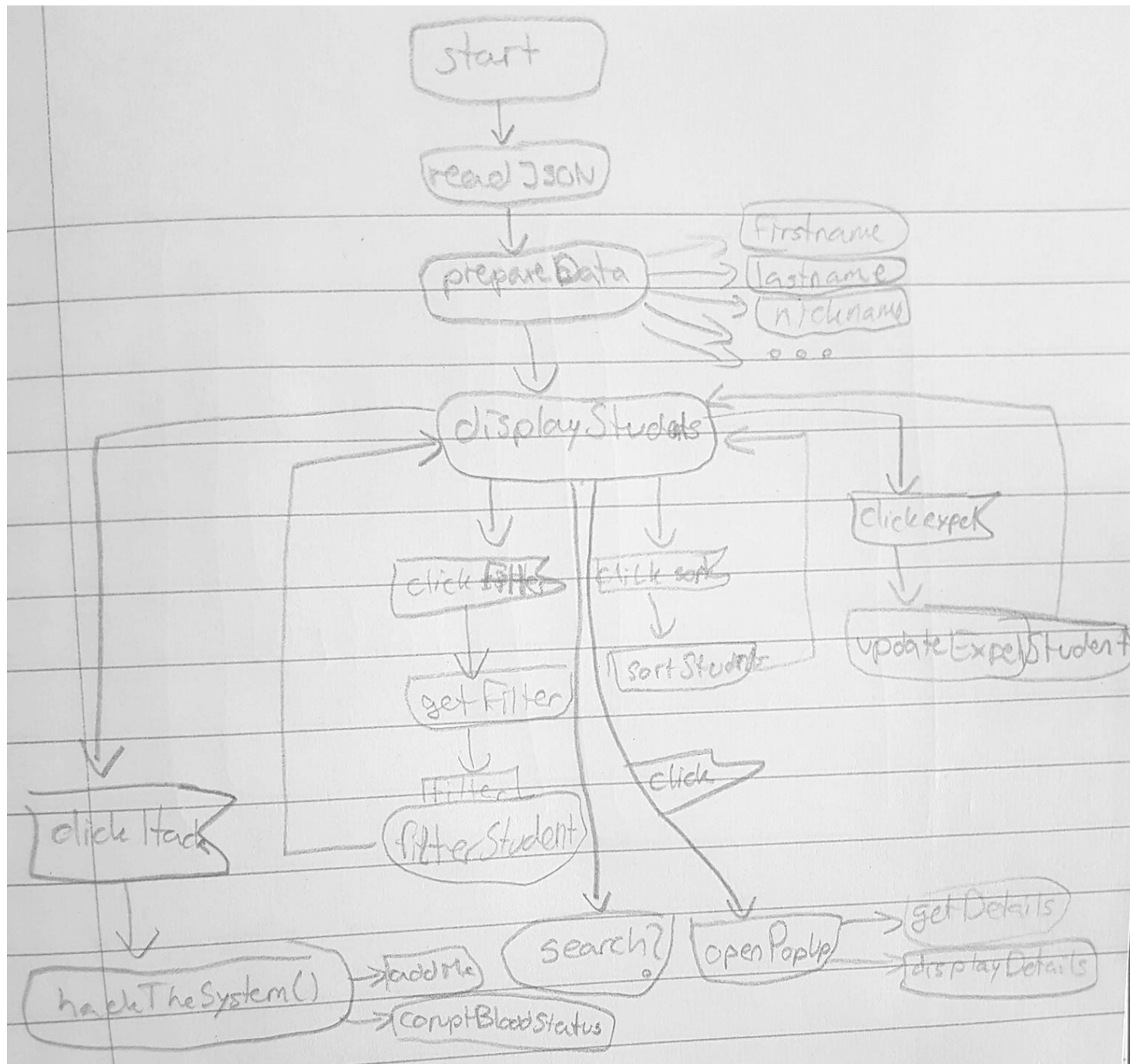
As an administrator, I want to be able to only appoint two students from each house a prefect, so that I don't accidentally appoint too many.

As an administrator, I want to be able to appoint certain students to the inquisitorial squad so that I always know who I have snooping around the school.

Overview of components:

<i>Components</i>	<i>Can/can't at beginning</i>
Listview	Can
Popup with details	Can
Filtering list	Maybe
Sort list	Maybe
Expel students	Maybe
Show info numbers on top of page	Can
Prefect	Can't
Inquisitorial squad	Can
Blood status	Can't
Search	Can't
Hacking	Can't

First version of activity diagram (before coding everything):



Expelling - pseudocode:

When an “Expel” button is clicked, call the function `expelStudent()`, unless `student.firstname == Sarah` and `student.lastname == Frederiksen`. If that is the case, then clicking the button just results in the text to change.

Firstly, find the student’s place in the `allStudents` array by for-looping over it looking for the `id`. Set `student.expelled` to `true`. Remove student from being a prefect and in the inquisitorial squad.

Then add to `expelledStudentsArray` by pushing. And remove student from `allStudents` array with `splice`.

Lastly call `buildStudentList()` to display list that now misses the expelled student.

This all translates to:

```
if (student.firstname === "Sarah" && student.lastname === "Frederiksen") {
  clone.querySelector(".expell").addEventListener("click", (student) => {
    document.querySelector("#Sarah .expell").innerHTML = "YOU CAN'T EXPEL ME";
    document.querySelector("#Sarah .expell").style.color = "red";
  });
} else {
  clone.querySelector(".expell").addEventListener("click", expelStudent);
}

function expelStudent() {
  //find student's place in array
  let thisStudetIndex;
  for (let i = 0; i < allStudents.length; i++) {
    if (allStudents[i].firstname === this.parentElement.parentElement.id) {
      thisStudetIndex = i;
    }
  }
  //set to expelled and remove from prefect and inq squad
  allStudents[thisStudetIndex].expelled = true;
  allStudents[thisStudetIndex].prefect = false;
  allStudents[thisStudetIndex].inquisitorial = false;

  //add to expelled students array
  expelledStudentsArray.push(allStudents[thisStudetIndex]);
}
```

```

//remove from allStudents array
allStudents.splice(thisStudentsIndex, 1);
studentsOnDisplay = allStudents;

//build list
buildStudentList();
}

```

Bloodstatus - pseudocode:

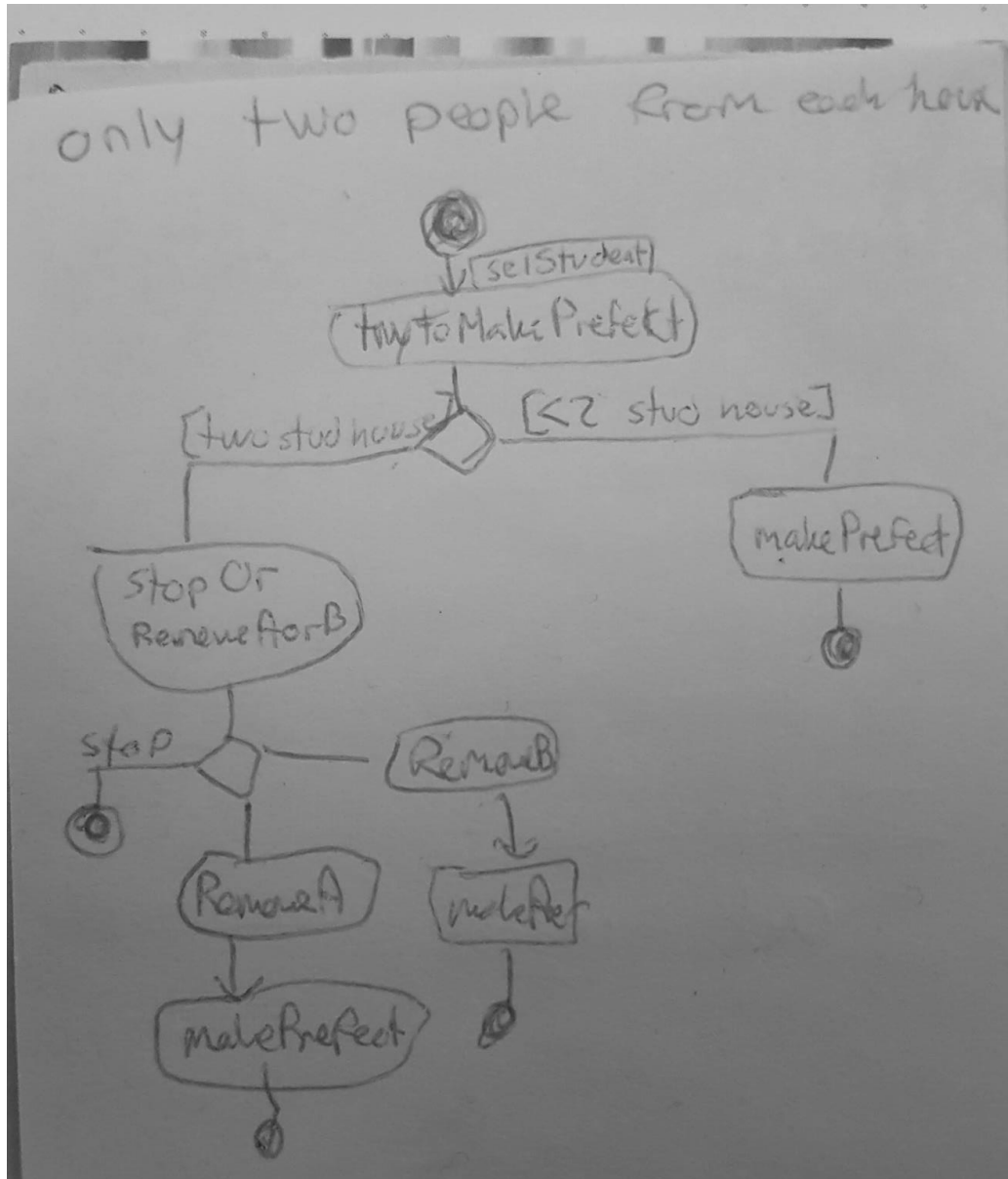
When the DOM loads start() is called. Firstly the JSON with the student info is loaded which also calls for the creation of all the students arrays. So when allStudents !== [] (when the students array isn't empty), the lastname JSON file is loaded. That calls on the function setBloodStatus(JSON) which sets student.bloodstatus. If their last name is in the "half" array then it sets it so "half-blood". If they're in the "pure" array it sets it to "pure blood". Else it sets it to muggle born.

```

function setBloodStatus(json) {
  allStudents.forEach((student) => {
    if (json.half.includes(student.lastname)) {
      student.bloodstatus = "half blood";
    } else if (json.pure.includes(student.lastname)) {
      student.bloodstatus = "pure blood";
    } else {
      student.bloodstatus = "muggleborn";
    }
  });
}

```

Prefects – activity diagram:



Student prototype:

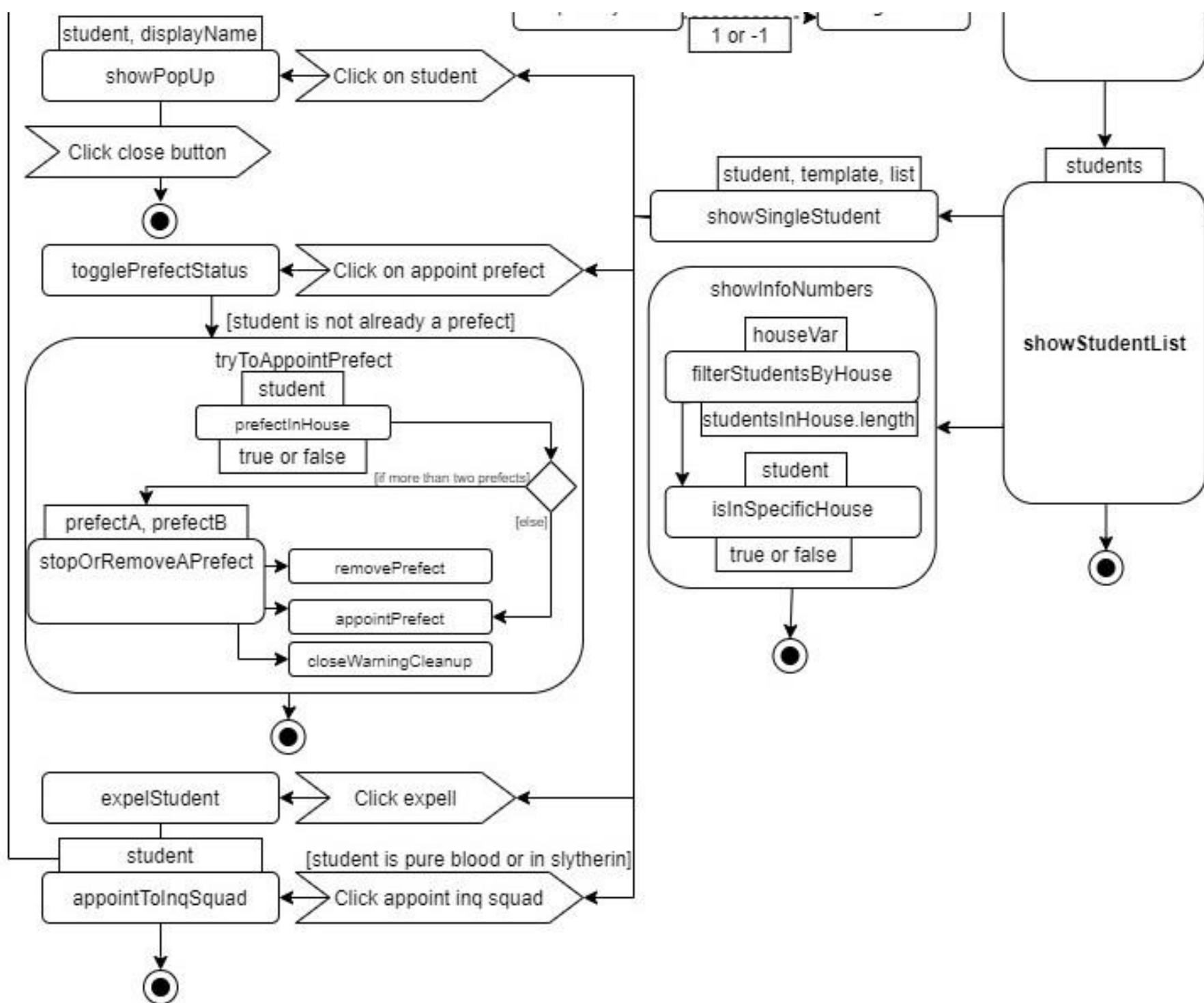
```
//prototype for all students
const Student = {
  firstname: "",
  middlename: "null",
  lastname: "",
  nickname: "",
  image: "",
  house: "",
  prefect: false,
  expelled: false,
  inquisitorial: false,
  bloodstatus: undefined,
};
```

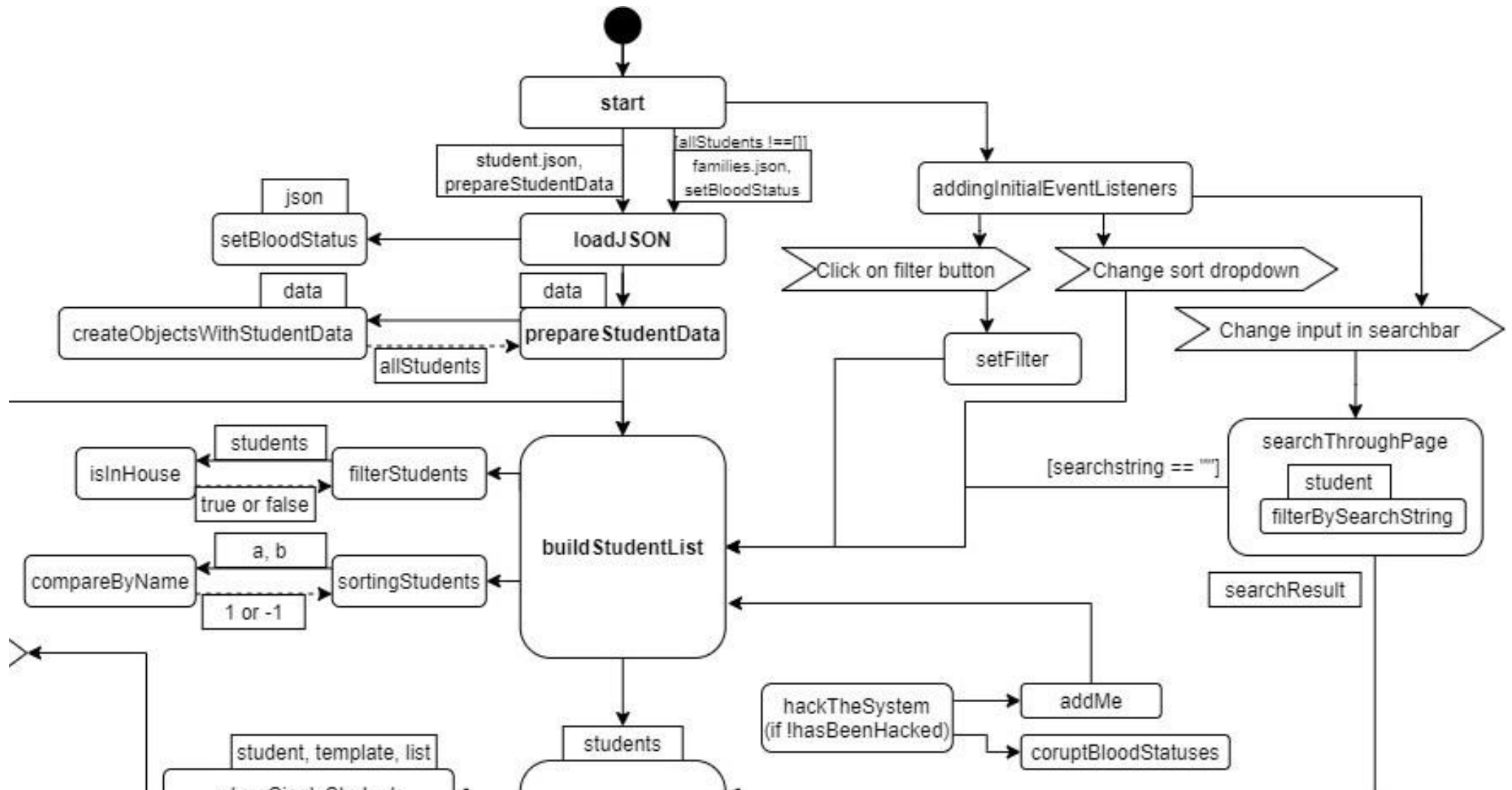
Complete activity diagram

Since the text is a bit small I've pasted some zoomed in closer looks at sections on the following pages.

The complete diagram can also be viewed here: http://sarahfrederiksen.dk/kea/frontend/hacked_hogwarts/activity_diagram.jpg

The system can be hacked by using `hackTheSystem()`; in the console or by clicking the first 's' in 'students' in the h1 "Hogwarts students list".





That's **the end**, thanks for reading :)