CS3099 Deliverable 1

Group 24

General Functionality

Written by 200007413, 200032853 & 190006961

Required Functionality

- Web server accessible from the St. Andrews local network
- Account creation / login
    - Users can sign up/login on our server
    - Users signed up on other federation sites (20's supergroup sites) can login to our website
    - Users signed up on our website can login into the other federation sites
    - Users' password will be encrypted when sent to server to ensure privacy
- Account management
    - Users can hold different roles, giving certain permissions, such as admin, solver, checker
    - Users from the other supergroup sites can hold different roles than they held on their original server.
- Playing different sudoku layouts/variants
    - Selectable list formed by layouts created by users signed up on the server, as well as from the other supergroup severs
    - The ability to save progress for the current puzzle being played when logged in
- The ability to create new sudoku layouts on our server
    - Puzzles created shall be stored on our server and will be playable on sites in our federation

Further Functionality

- Sudoku creation
    - Solvability checking of a sudoku for setters by the system before publishing, to make sure that puzzles are uniquely solvable
    - Set difficulty for newly created sudokus
    - Save drafts of new sudoku creations
- Solver aids
    - A pencil/note system for solvers. Where boxes without a number assigned can have the numbers 1-9 put small in the corners, edges, and center of the box to demarcate potential solutions for that box.

    - Some puzzle hint choices. To highlight all the boxes of a specific number for clearer view, get reminders for number contradictions, and get answer hints when stuck.
    - A visual representation of when a user places numbers that create a contradiction, but not whether either number is specifically correct; i.e. two 9's in the same row
    - Save progress of certain puzzles or pause the game
- Personalisation of account
    - A puzzle recommendation system, where users have personalised puzzle variant suggestions
    - The ability to add information about yourself on your profile, including a profile picture
    - The ability to add friends or follow creators, and see the puzzles they have played, rated, and solved, or get notified for new puzzle creations
- Puzzle-oriented
    - The ability to see statistics for average time-taken to complete, solving rate, user speed rankings, and reviews for each puzzle
    - The ability to rate or comment on a puzzle.
- Site-oriented
    - Feedback mechanism for all users to aid in maintenance focused around the users of the site for bugs, site improvement ideas, and QnAs.
    - Search bars for certain user accounts
    - A way of displaying profiles of those who have accounts on other sites but have played games on this site
    - Admins can control the account rights of any site users

Technologies to be used

Written by 200032853 & 200007413

Frontend

We decided to use the VueJS framework for creating the web frontend. In our aim to keep our code as minimal and efficient as possible for reliability and maintenance purposes, we considered avoiding javascript for the frontend. However, the website's primary function is to allow users to play sudoku, and so we decided the added responsiveness javascript offers over pure HTML was worth the added complexity.

We decided to use the VueJS framework for our frontend. This helps standardise our javascript code, as VueJS comes with vast documentation on best practices for certain elements. This is especially important for the interaction with the backend, as VueJs makes adapting to changes in format of the puzzle configuration JSON data relatively easy. We picked VueJs over other

frameworks, such as ReactJs, as some members of the team have more experience with VueJs, and it is more intuitive than the design language of React. It also has great official support for many features we may require, and is well documented.

Along with VueJs, we decided to implement the stylesheet preprocessor Sass. This aids in organising the source code for styling, as well as automatically adding cross-browser rules to help keep the style appear consistent across browsers. Again, we intend to keep a modular folder structure, loosely based upon the guidelines from Kitty Giraudel (see. https://sass-guidelin.es/).

Backend – Server

For the site, we require a web server open to the local St. Andrews network. We decided to use NGINX as the server engine instead of others (e.g. Apache) for many reasons: firstly, the St. Andrews linux host servers are already setup for easy NGINX configuration; our team also has more experience with NGINX, which will save initial time and effort.

Backend - Language & API Comparison

We decided to use javascript to create our backend, implementing express.js with NodeJs. Our initial list of considerations were Java, Python, and JavaScript. Despite all team members having the most experience in Java, and its excellent reliability, we lack experience of using Java for web development. Moreover, Java requires an added step of compilation before running which is both time and resource intensive compared to the alternatives.

Being left with javascript and/or python, our supervisor suggested Django as a potential technology which is built in Python. It has the benefit of being very stable, with it's MVT (Model-View-Template) structure, as well as having a full user authentication library, making user signup/login system, permissions/roles, accessibility, and session-management implementations relatively simple, and vastly more secure than building up from scratch. However, not all of us have prior experience in using python on the backend.

Alternatively, we considered, and ultimately chose, nodeJs which enables the use of Javascript. NodeJs enables us to use a package called express.js, which focuses on creating APIs, which will be important for our site's session management, and interaction with the other sites in our federation.

Using javascript for the backend makes integration with the VueJs frontend easier as both are built with javascript. In addition to the knowledge and experience in nodejs and express some team members have, unifying the backend and frontend languages means the skills gained from learning frontend are more applicable to the backend, which is important as we do not have enough members for dedicated frontend/backend developers. However, this similarity may cause confusion as Vue and Node do to have some particular differences in features and code style. It was also noted that nodejs is slightly less secure than Django in many cases, as Django

is more rigid in how it is used, thus reducing the chance for misconfiguration, which is especially important for our project as we will be accepting a lot of user interaction/input which always poses security threats, requiring resources to manage and mitigate.

Backend - Database & Language/API decision

As we are going to need to store data on users, roles, permissions, along with information about the other supergroups, and different puzzle layouts, we require some form of database. The two main options we considered are going for a standard mySQL relational database, or using MongoDB, which is a noSQL database.

We decided that a noSQL database is more in line with the functionality we are looking for in our database. This is because mySQL databases have a very strictly defined schema, and so any changes made to the types of information stored in a table is much more difficult to implement, and as we are using the AGILE development method, MongoDB provides the flexibility we require. Moreover, MongoDB stores data in the format of BSON, which is a binary representation of JSON, allowing easy retrieval and storage of information in JSON too. As interserver (within supergroup) and server-client communication shall be conducted in JSON format, MongoDB will save time for data structure conversion for any AJAX actions.

Due to our decision on using MongoDB, Nodejs and Express in Javascript will be used in backend development as they support MongoDB as a backend database, with simple codes to build connection to the database. It would be a more ideal option than Django, which mainly supports SQL databases.

Cookies usage

The use of cookies in our website will be mostly limited to session management for logins. As most information related to the puzzles is stored on-server, for access across devices, cookies are only really needed to remember the current login-session for convenience, and will integrate with the backend authentication system.

Scrum

Written by 200036815 & 200007047

Methodology

Role distribution

In an incremental scrum process, as a group of 5, we have :

1) A product owner, to set a clear direction for the team, by collecting the needs of customers with user stories, and setting them into the product backlog, in descending order of their priorities;

2) A scrum master, to monitor and facilitate the progress, ensure tasks are done well, and glue the team together;
3) Developers, to maintain the sprint backlog, by determining the goal of a certain sprint, extracting items from product backlog, splitting them into tasks, and conduct concrete design and develop process;
4) Testers, to find out the defects or bugs in the product, verify and validate that user needs are met fully and appropriately.

Roles will be changed after some sprints according to our discussion in the first scrum meeting.

Scrum process and sprint cycles

According to the lecture slides, a complete scrum process would consist of planning, the sprints, and closure. During the outline planning, we drew a big picture of what our product will be like, set general objectives and determined role distribution. In each sprint (2 weeks), we develop the system incrementally following the sprint backlog, evaluate retrospectively in scrum meetings(twice a week, Tue. & Thu.) before proceeding to the next increment. At the end of the project, we will reach the project closure phase with every item in the product backlog removed and accomplished, and will wrap up the project and deliver, produce all documentation required and collect lessons learned. The project is closed then.

Tools for Scrum

We set up a group in the Microsoft planner through SharePoint under the site of “cs3099-g24”. We use kanban to store the product backlog, and each sprint backlog per cycle, and use OneNote to write meeting minutes.

We also have a file directory where all of us can edit and add files, to share and archive files we create, such as the chart we create at first to collect everyone’s experiences and preferences on web and sudoku.

First Scrum meeting & reflection

Priorities

Our tasks for the first sprint are grouped into 2 categories, back-end and front-end. For the front-end, we planned to design the interface for user authentication, write a html for displaying the interactable puzzles and allowing users to create their puzzles. For the back-end, we aimed to enable user authentication and set up the server and the databases.

Roles

For the first sprint, 200036815 will be the product owner, 200007047 will be the Scrum master, and all of us will be developers and testers as there are only five members. We are planning to change our roles next sprint so that everyone is able to take part in different roles of the Scrum. In terms of roles in tasks, 200032853 and 190006961 will focus on the front-end tasks. For the first scrum, this will include a very minimal framework, focusing on the functionality of rendering a list of games to be filled once we have puzzles setup, being able to render the puzzles themselves, and being able to submit authentication forms for signup/login. 200007413,

200036815 and 200007047 will focus on backend tasks, which will include setting up the base NGINX server, and databases for puzzle configurations and user logins/roles/permissions.

Progress of Scrum

We finished the phase of outline planning at the end of week 4, allocated the roles we will be in the process, and have the first deliverable out. We also decided that the length of a sprint in our scrum process is two weeks, starting from week 5 this semester. Therefore, after the second sprint (at the end of week 8) we will have most of the tasks we need for the minimal viable product done, then for next sprints, we will fix bugs and defects, be able to hand in the first version of product in week 9 and continue to strengthen the functionalities. The last sprint of the semester would be the fourth one, ending in week 12, while the closure of the project would be at the end of the academic year.

Meetings

We are going to have standup scrum meetings twice a week (Tue.14:00 & Thu.10:00), either online on discord or in person in labs, to report what our progress is in a certain sprint to monitor the work.

Communication within supergroup

Written by 200007047

Meetings and communication

We planned to have a meeting in person every Monday. Each group will send 1 or 2 representatives to the meeting to discuss how the federation works together. The representative(s) will explain it to their group and work on the updates together.
We have created a discord server so that everyone in the supergroup can voice out anytime. There are also 3 channels for the protocols authentication, the user data protocols and the puzzle protocols. Therefore, we can find the format we need from the specified channels.

Federation and protocols from the meetings

Each federation will store its own user database and puzzle database. The communication protocol with the supergroup will be using AJAX and JSON to send login/ user data. If a server is requesting for user data, it will send the email and password to the server with that data entry. The server holding the user data will check if that user exists and if the password is valid and send back some flag indicator. If the user exists and the password is valid, the server holding the user data will send back the data agreed by all federations. Currently, the user data that will be sent back includes the email, name, username with Uids in the format G24-17 implies user 17 of group 24. We should be able to convert the database into the agreed JSON format before sending it to other groups. We will include names and IPs/ URLs of all federations into a server table/ list stored in a text document.
A federation will store and use its own puzzle database, so the server which owns the puzzle database will store the game progress instead of the server which owns the user data.

Initial Set of User Stories & Backlog

Written by 200032853, 200036815 & 190006961

User Stories

User stories highlighted in grey are to be confirmed

'All users'
As a user, I want to be able to log in to my account with a password
As a user, I want to be able to set up 2FA to protect my account incase my password is found
As a user, I want the site to remember my account username such that I do not have to type my full username
As a user, I want to receive emails when a login on a new device is detected, to help alert me to unauthorised access to me account
As a user, I want to edit my user profile.

'Site user'
- Indicating all types of website users, including both solver and setter (everyone except the admin)
As a site user, I want to be able to register a new account with email
As a site user, i want to be able to login with my username and email
As a site user, I want to choose my role on the site (solver/setter/both)
As a site user, I want to raise questions for the admin when I have difficulties in the functionality of this website.
As a site user, I want to be able to provide web-developers suggestions for site improvements
As a site user, I want to be able to report bugs.
As a site user, I want to be able to search for other user accounts such that I can see how well others play/ what puzzles they have made.
As a site user, I want to be able to report inappropriate comments I see on puzzles.

'Administrator'
As an administrator, I want to view the list of site users, their roles and rankings such that I know who uses the site.
As an administrator, I want to receive feedback on the site operation from users.
As an administrator, I want to have the right to ban certain users to prevent inappropriate behaviours / speeches on the site.
As an administrator, I want to detect inappropriate languages used in feedback for puzzles such that no users shall feel offended by other users.
As an administrator, I want to check that every new sudokus created on my site is original such that there will not be copying issues among setters.

'Solver'
As a solver, I want to see how many times I have played on the site.

As a solver, I want to see the percentage of my success in finishing the puzzle such that I can know if I did well.
As a solver, I want to see how many times I have played on a specific sudoku.
As a solver, I want to be able to select a game of sudoku from the local server by difficulty, popularity, creator, time created etc.
As a solver, I want to see the number of people who have played a game and who have solved it.
As a solver, I want to be able to select a game of sudoku from other servers.
As a solver, I want to see the difficulty level of a certain sudoku.
As a solver, I want to set a number per box.
As a solver, I want to delete the number I put in a box.
As a solver, I want to have a timer so that I can see how fast I played.
As a solver, I want to pause the timer so that I can have a break during solving.
As a solver, I want there to be an indication of how many times I paused while solving alongside the time I solved a puzzle in.
As a solver, I want to have an optional automated checker so that I know whether I put in the correct number.
As a solver, I want to save the game halfway so that I can resume playing next time.
As a solver, I want to get hints or restart the game when I am stuck.
As a solver, I want to be able to start another new game while saving the current game progress when I am stuck.
As a solver, I want to be able to highlight the rows and columns of the selected cell and also the 3x3 box I am currently in, so that I can check the restrictions before filling in the number
As a solver, I want to be able to highlight all the occurrences of a certain number when the cell with that number is chosen, so that I know where the remaining(s) can be.
As a solver, I want to leave feedback on the setting of the sudoku.
As a solver, I want to rate the sudoku in terms of difficulty and whether it is fun to play.
As a solver, I want to follow certain setter users who make good puzzles such that I can be reminded when they release a new sudoku.
As a solver, I want to view my server ranking in terms of speed of solving a sudoku.

'Setter'
As a setter, I want to put the number in a box.
As a setter, I want to get a warning when the number I put in clashes with the other in its row/ column/ 3x3 box.
As a setter, I want to save the soduku I am currently setting so that I can resume setting next time.
As a setter, I want to determine the difficulty mode of the soduku I created.
As a setter, I want to change the colour of my font or background.
As a setter, I want the site to check if my sudoku is valid such that the published sudoku is guaranteed to be solvable and have one and only one solution.
As a setter, I want to be able to provide correct answers so that solvers can check correctness.
As a setter, I want to leave feedback on the operations of setting the sudoku.

As a setter, I want to create sudoku variant puzzles, like using boxes in different shapes instead of squares.
As a setter, I want to be able to view a feed of all comments left on my puzzles.

Initial set of Backlog

Puzzle display & interactivity
As a solver, I can view a basic display of an interactive sudoku puzzle.
As a solver, I can choose which number to put into an empty box.
As a solver, I can remove a number from a box I filled when I am unsure about it.
As a setter, I can have an empty 9x9 box template for my sudoku creation.
As a setter, I can add and delete numbers 1-9 into any box.
As a setter, I can have my creation saved in the site database and be viewed by players.

User authentication
As a site user, I can register my own account such that I can keep track of my sudoku records.
As a site user, I can have a username and the password for my account.
As a site user, I can choose my role in the site.
As a site user, I can login to my account with username and password such that my account can be secured.
As a web administrator, I can login to manage the rights of each user and view all user information.
As a web administrator, I can make sure all usernames on my servers are unique, so login requests always return the correct user.

Overview of plan in semester 1

Written by 20007413, 200032853 & 190006961

With our site, we plan on creating a site where users can play and create sudoku puzzles with many different variants. Our aim is to stand out with a clean design, and a better way of finding new layouts/variants which the player will enjoy. We also have various ideas of extra user stories which can improve user experience and interactivity with the site, such as validity checks, feedback, ratings, customised search, and a dashboard for the time used in solving, etc.

In the first semester, we would like to prioritise the basic functionalities along with some common sudoku functions for our site. Including the tasks in our first sprint, our minimum viable products shall allow users to register and log in to their own account of certain picked roles with guaranteed security. The system should also allow login of users registered on other federation sites. Setters should be able to access an easy-to-use interface to create their puzzles to be stored in the backend databases with the owners indicated. Solvers should be able to play the

puzzle from databases, possibly with a pencil-note function too. There should hopefully be some rudimentary form of user profile containing information such as username and previously solved puzzles. The method and structure of which the sudokus and account information will be stored in MongoDB shall be decided in the first stage, but open to minor changes in the next semester.

The implementation of other extended functionalities will be mostly conducted in semester 2, such as automated solvers, answer checkers, comments, and other complex features. However, it is possible that we will include some more in case of good progress of the minimum viable products, which will be brought out in our weekly meetings.