***420-K30-HR – Assignment 1 Systems Maintenance Ramp Up***

Date Assigned: Thursday September 5th, 2024

Date due: **Friday September 13th @ 11:59PM in Moodle. See notes below re interim delivery dates.**

You will be given time during class on **September 5th, 10th, 12th** to work on this assignment. Each class will begin with a short overview of the topic for the day and assignment expectations.

Most of this assignment is individual, but you may work with your teammates while establishing your setup for Part A. Part C involves group work, but everyone must understand branching and merging and learn how to branch and deploy on their own. Also, although you work on things as a group, everyone must provide answers to all sections in their individual submissions.

**Part A is due at the end of class on September 5th.**

**Part B is due at the end of class on September 10th.**

**Part C is due at the end of class on September 12th.**

**Part D is due as the final submission September 13th@ 11:59PM.**

**No assignment submissions will be accepted after the final submission date, i.e., nothing submitted after Friday September 13th @ 11:59PM in Moodle will be accepted.**

**If earlier parts are submitted late, the usual Computer Science late policy (as per the course outline) applies up to the cut-off date of the 13th. However, missing interim delivery deadlines is strongly discouraged.**

Note: If you have completed the part to submit on that day early, work ahead on individual parts.

**Learning Objectives**

Upon successful completion of this assignment, the student will be able to:

* Understand the maintenance requirements for their assigned system;
* Understand the high-level architecture of the system;
* Set up the development environment for the project;
* Understand how to build and run the project they will be working on;
* Do exploratory testing on the system;
* Identify and report on initial problems found with the system;
* Analyze backlog items;
* Understand how to deploy the system; and
* Come up with an initial list of critical items to address in the maintenance project.

**Fall 2024 Projects, Teams, and Customers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | **Team** | **Azure Link\*** | **Customer** |
| **Nursing** | Sebastian  Jade  Jacob  Sabrina | <https://csazure.cegep-heritage.qc.ca:8080/F2023-DevProject/Project%203%20-%20Nursing%20App> | Cathy Dufour |
| **Library** | Matteo  Julia Maria  Heraj  Cristian  Christopher | <https://csazure.cegep-heritage.qc.ca:8080/F2023-DevProject/Project%201%20-%20Library%20App> | Connor Spearman |
| **HFA** | Pax Bertin  Joshua  Brandon  Ryan  Claude | <https://csazure.cegep-heritage.qc.ca:8080/F2023-DevProject/Project%204%20-%20HFA> | Julien Fournier |
| **Fitness** | Pierre  Gautam  Benjamin  Parker  Carlos Branimir | <https://csazure.cegep-heritage.qc.ca:8080/F2023-DevProject/Project%202%20-%20Fitness%20App> | Jillian Clarke |

Link to existing AMS on csdev:

<https://csdev.cegep-heritage.qc.ca/Projects/AMS/>

Information about test accounts to use during development will be provided in a separate file on Moodle.

To Start:

1. Make sure you read the **entire assignment** to understand the scope of work.
2. At the beginning of any maintenance project there is a discovery phase where you ramp up on the project and discover what the current artifacts of the system are and what state they are in. Your task in Assignment 1 is to become familiar with your project, understand what it is about, and determine which area(s) you think we should concentrate on for maintenance this term. This will prepare you and your team for meetings with the customer later in the term.
3. Opening status summary:
   1. These projects were all developed from scratch last year by the Dev Projects teams.
   2. None of them are in production yet.
4. Note that you may feel overwhelmed trying to ramp up on unknown projects. This is a normal reaction and why we do a discovery and planning phase early in the maintenance cycle to ensure everyone is established and ready to contribute. There may be many hurdles to getting the systems up and running and becoming familiar with its current state.

**To Do:**

## **Locally Install and Discover the System (Individual)**

For your response to Part A, rename this document to **YourUserName\_K30\_A01\_PartA.doc** and submit it to Moodle in the appropriate submission folder.

Provide the information/ answers in the appropriate section of this document.

**The goal for Part A is to understand the environment.**  To do this you will need to:

* Fetch the code for your project from Azure
* Get it running locally on your computer (i.e., using VS 2022) using the project Run Book as your guide;
* Run the code to make sure you’ve got all the elements needed to run the application; and
* Acquaint yourself with the system and artifacts.

*Note find a copy of the latest runbook for project. If you find that the information you need is missing from your run book, look at one of the other ones for clues. Often the ones that have been developed or maintained recently, have better/more up-to-date instructions.*

* 1. **Identify current state of collateral**

Complete the table below, updating the current state (available/not available, complete/not complete), location (i.e., where in Azure to find artifacts – this should be a link), and any concerns you have with the information so far. Note that some rows may have multiple line items.

|  |  |  |  |
| --- | --- | --- | --- |
| **Artifacts** | **State** | **Location** | **Concerns** |
| Installation/Runbook | Available, completed | Overview -> Summary | Some steps include AMS, which is not accessible for us |
| Design Documentation | Available, completed | Overview -> Summary | NCET currently relies on AMS to do authentication and authorization |
| Azure Project specific home page[[1]](#footnote-1) | Available, completed | Overview -> Summary |  |
| Software (Azure Code/Trunk/branches)[[2]](#footnote-2) | Available, completed | Overview -> Summary |  |
| Database, Other technical artifacts | Not available, completed |  | AMS is currently not accessible |
| Azure Work items[[3]](#footnote-3) | Available, not complete | Sprints | Tasks are open for future sprints and tasks are unassigned in previous sprints |

* 1. **Your Setup**
     1. Setup your Azure and Visual Studio to access the latest source code. Start by using the latest branch available**, but make sure you do not check in any changes.**
     2. Each student must have their own database. Use SSMS to login to the SQL Server DB on csdev using **Windows Authentication** to create your own copy. Remember to change the connection string in your local project to point to your **own** database copy. NEVER check in your connection string change into the maintenance release. Leave the maintenance release working from a common database. **For the AMS team it is especially important that you do not make any changes to the main AMS database that is being used by everyone else!**
     3. Make sure that you can run your project.
     4. **Fill in the table below. Put a yes/no indication in the cell for each completed task and ensure that you demonstrate this to the professor before the end of the work period on September 5th.**

|  |  |
| --- | --- |
|  | Completed? |
| Personal copy of source code | yes |
| Personal copy of database | yes |
| Verified personal environment, up and running | yes |
| Demonstrated to professor | yes |

* 1. **Answer the following questions for your assigned project:**
     1. Did you get your personal instance of the application up and running smoothly? List any difficulties (technical or interpersonal team dynamics) you ran into. Identify the changes needed to the Run Book for your project that would help this process.

I did get my personal instance of the application up and running smoothy. There were technical difficulties considering AMS is currently not working, so we had to bypass the use of the system with temporary users to be able to access the system with different permissions.

* + 1. In your own words, what is the system used for? Provide a couple of paragraphs to describe the system and provide an overview of the key requirements for the system.

The system is a tool that allows coordinators, teachers, and students to do evaluation-related tasks.

Coordinators can create courses and start them, assigning a lead, start date, and evaluation due dates. They can also update the courses and change the competencies and criteria.

Teachers can evaluate their students, and create criteria for their evaluations, and

Students can evaluate themselves and view their teachers’ evaluations. They can also enrol themselves into courses with codes provided by the teachers.

This system can be used to have as a virtual assessment tool to minimize the amount of physical work handed to and from students to help keep things organized for all in the nursing program. It provides a centralized location where nursing staff and students can evaluate and communicate one another without the need of having paper copies and physical databases of these evaluations.

## **Initial Understanding of the System (Individual)**

For your response to Part B, rename this document to **YourUserName\_K30\_A01\_PartB.doc** and submit it to Moodle in the appropriate submission folder. Provide the information/ answers in the appropriate section of this document.

**The goal for Part B is to understand the more about the system.** To do this you will need to:

* Understand the application’s purpose and basic functionalities;
* Understand the workflow of the application; and
* Understand the types of users and roles and how they differ.
  1. **Users and roles:**

Using the repository artifacts you found and read through in Part A, identify the user/roles for your system. Identify the test account(s) to use for this role and ensure you can login. Enter this information in the table below. Add rows as necessary.

| User Role | Summarize how they use the system | Test account login | Who shows up as this test user on csdev? | Comments |
| --- | --- | --- | --- | --- |
| Nursing Coordinator | Can create and manage courses, assign instructors to courses, evaluate students, set criteria and competencies. Highest level of power within the system | NC | Coor Dinator |  |
| Nursing Student | Lowest level of power within the system. Can self evaluate and see instructors’ evaluations on them. | NS | Stu Dent |  |
| Nursing Instructor | Can evaluate students and see students’ evaluations. | NI | Instru Ctor |  |
| Nursing Lead | Can manage existing courses but cannot create them, Can also do everything the Instructor can | NL | Lead Er |  |

* 1. **Workflow and functionality**

Later in the term we will learn about different test methods. One way to gain an initial understanding of a system is to use exploratory testing, where testers interact with the application to explore the application’s functionality in an ad-hoc manner.

For each of the roles, logon and use the application to explore the workflow and functionality.

* + 1. Explain the workflow of the application and its basic functionalities for each role.

Nursing student: Can view their courses as well as self evaluate and see the teacher’s evaluation for that course.

Nursing Instructor: Assigned to one course, no option to view or manage other courses or manage the course they are in. Can evaluate a student for that course or see the student’s self evaluation.

Nursing Lead: Can do everything the instructor can but can also be assigned to multiple courses and manage courses that they are assigned to.

Nursing Coordinator: Can do everything the lead can but can also create courses and assign teachers to those courses.

* + 1. Identify any problems you found/ ran into. For each issue, investigate the existing backlog of user stories, tasks, and bugs to see if your issue is already known. Record your findings in the table below. Add rows as needed.

| Issue Description | Login you were using | Known issue?  (enter either yes or no, and if yes, include Azure story, task, or bug ID) | Comments |
| --- | --- | --- | --- |
| Hacky solution for login | All | Yes, 2150 |  |
| Adding 26 or more criteria starts using ASCII code characters | NL, NC | Yes, 2004 |  |
| Potentially if 2 or more students or instructors had the same first name and last name you would not be able to differentiate | All | Yes, 1878 |  |
| Overall comments vs General Comments | All | No | Not specific what the difference between these is |
| Add dark mode | All | Yes, 2134 |  |
| Ability to hide evaluation from student | All | Yes, 2136 |  |
| Allow exports to pdf/image | All | Yes, 2138 |  |
| French translations | All | Yes, 2135 |  |
| Courses cannot be deleted | NC | No | Done for legacy reasons, but may become an issue |

## **Deployment and Additional Setup (Individual and Group)**

For your response to Part C, rename this document to **YourUserName\_K30\_A01\_PartC.doc** and submit it to Moodle in the appropriate submission folder. Provide the information/answers in the appropriate section of this document.

* 1. **Azure Homepage/Readme (Individual)**

All CS projects must include an Azure home page with a README that provides context such as:

* What is It?
* Documentation – pointer/link to where the doc is located and how to find out more
* History & Milestones – i.e., when developed, put into production, and other relevant milestones.

[Microsoft Reference](https://docs.microsoft.com/en-us/azure/devops/repos/git/create-a-readme?view=tfs-2018) and [sample](https://github.com/apache/tomcat).

Ramp up on README best practices and identify any updates required for your project README file. List them here:

* Should elaborate on user roles, describing what they do
* Getting started should include steps for developers to run the project locally
* For build and test and system test and coding standards, instead of just pointing to the documents, there should be a brief version directly in the readme
* The history section should include dates or more detail on development milestones
  1. **Azure Project Documentation (Individual)**

Consider if all the necessary documentation is in Azure or if important information appears to be missing. Identify any additional documentation to add to your repository for the next maintenance cycle. List it here:

A detailed history section is missing, but the rest of the documentation is there.

* 1. **Versioning (Individual)**

All projects in this maintenance course will need to upgrade their project version to the next major version following the [Microsoft Versioning standards.](https://docs.microsoft.com/en-us/javascript/api/sp-core-library/version?view=sp-typescript-latest)

* Find where the version of the project is located, both in the code and on the homepage or navigation menu (if the version is shown to the user).
* If the version is not shown to the user, please identify this as an issue.

Current version (write it, and show a snapshot from your local copy below):

A screenshot of a computer

Description automatically generated

Version 1.0.0

Where in the source code is this done (identify the file name and show relevant code below):

    private readonly IEnumerable<ReleaseInfo> \_releaseInfos = new List<ReleaseInfo> {

        new ReleaseInfo {

            SemanticVersion = "1.0.0-rc-3",

            ReleaseDate = new DateOnly(2024, 05, 2),

            Summary = "Bugs found and fixed between the second and third release candidate.",

            Changes = new List<String> {

                "Fixed 1984: Self-enrol breaks student home page because evaluations are not created by it",

                "Fixed 1980: Misleading message on the Instructor home page",

                "Fixed 1983: Success message for deleting a course run is not descriptive enough",

                "Fixed 1956: ReturnUrl is incorrect in ViewAllEvaluations",

                "Fixed 1980: Misleading message on the Instructor home page",

                "Fixed 1983: Success message for deleting a course run is not descriptive enough",

                "Fixed 1984: Self-enrol breaks student home page because evaluations are not created by it",

                "Fixed 1990: Enrolment code breaks if the course name has white space type characters in the name",

                "Fixed 1977: Overflow if words are too long when creating course competencies/criteria",

                "Fixed 1979: Participants Assignment confirmation checkbox is styled weirdly on Safari (MacOS)",

                "Fixed 1982: Images from the help page are outdated and do not show an exact representative of the system's state",

                "Fixed 1957: Update release notes",

                "Fixed 2040: Version information link in the footer doesn't reflect the reality",

                "Fixed 2041: Inconsistent Manage buttons in the Manage Courses page for leads ",

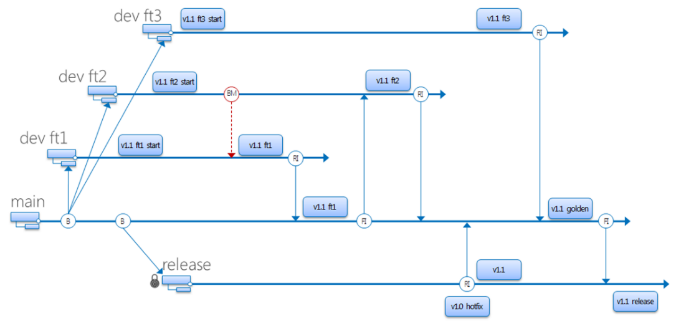
                "Fixed 2042: Submitting start/manage course run with an instructor or student user id works"

            }

        },

HomeController.cs

* 1. **Branching and Merging (Group and Individual)**



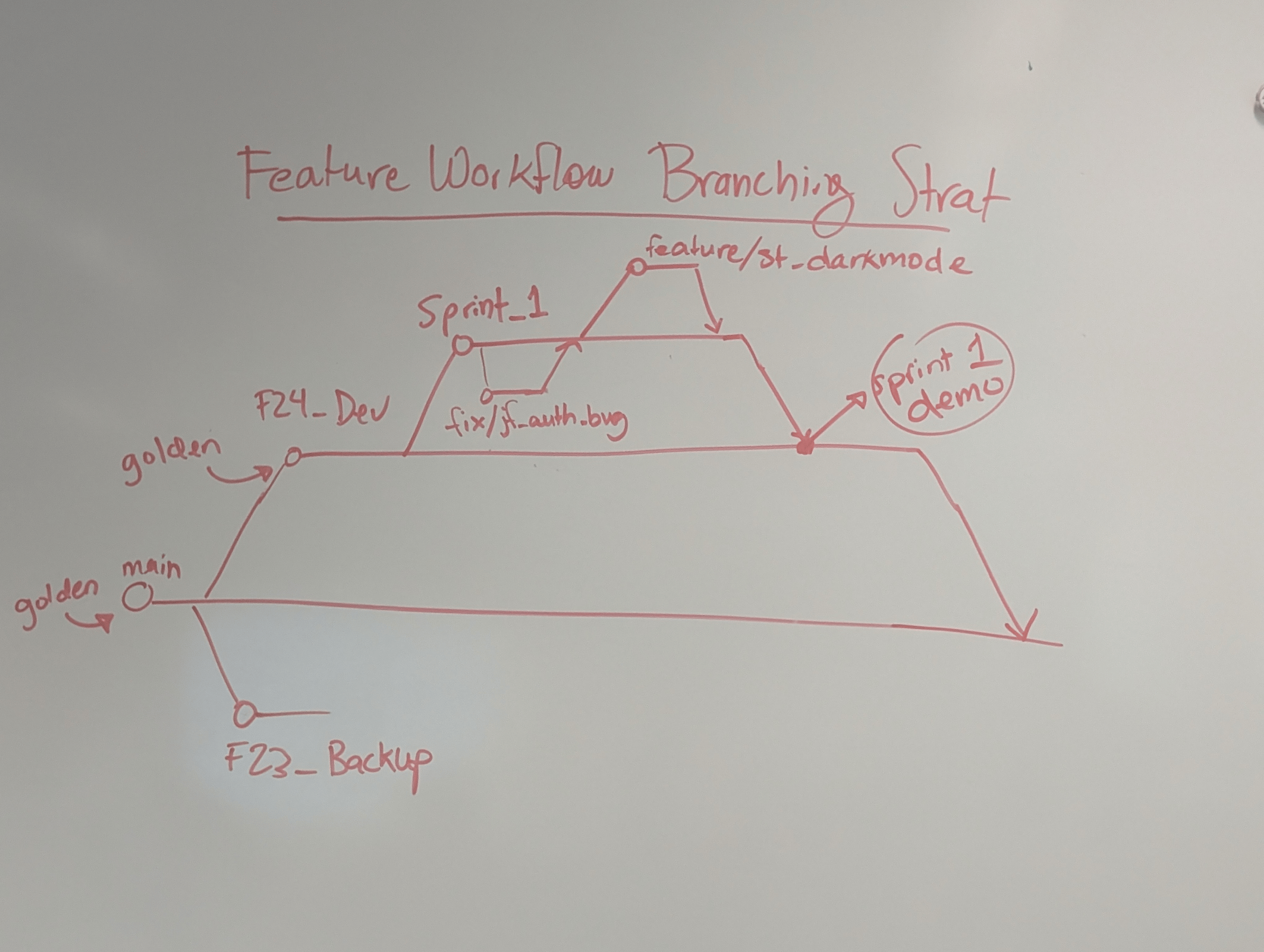
I suggest that if you need a refresher on Git branching strategies, you go through the following:

[Branch Strategies and Workflow](https://learn.microsoft.com/en-us/training/modules/manage-git-branches-workflows/).

Ramp up on Branching and Merging. **As a team**, create one new branch from the latest/main branch called *Project\_F24\_Dev*. We will use this as the team development branch for the term. Then, everyone should create their own development branch from that branch called …\_username. For example, for me it would be Nursing\_F24\_Dev\_RChan, etc.

Explain your branch strategy and show a snapshot of the branches below:

We’re going to have the golden F24\_Dev branch that we will merge into at the end of each sprint. During a sprint were going to make a branch and then make more branches off that sprint for our tasks and merge them until our sprint is over, that sprint is then merged and demo’d for the client at the end of the sprint with the new merges. At the end of the year, we will merge into main, but we will have a backup of the current state of the main to not upset last year’s 3rd years.



* 1. **Updating your setup (Individual)**

Update your Git and Visual Studio personal setup to access/use the source code from the personal development branch you created in Question 4. Confirm and demonstrate to the professor that this is working before the end of class.

Confirmed to be demonstrated

## **Understanding the Backlog (Individual)**

For your response to Part D, rename this document to YourUserName\_K30\_A01\_PartD.doc and submit it to Moodle. Provide the information/ answers in the appropriate section of this document.

The goal for Part D is to identify some critical maintenance work required for the system. To do this you will need to:

* Understand the application’s purpose and basic functionalities;
* Review the existing product backlog;
* Consider any additional issues you found during exploratory testing; and
* Come up with some recommendations for items the class should consider fixing, and why.

Note that the current product backlog has not been groomed, and does not have recent customer feedback, so the new requirements and user priorities are not reflected. Later in the term you will work on backlog grooming as a team, and you will gather requirements from the customer.

* 1. Identify 15 items you believe are important to fix this term. For each issue, identify the type of maintenance (refer to class notes), your assessment of priority, and if you have been able to reproduce or investigate it.

| Description of item | User roles involved | Known issue?  (enter either yes or no, and if yes include Azure story or bug ID) | Your assessment of priority and why | Able to reproduce?  (enter either yes or no, and if yes note how) | Maint. Type |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Marking Scheme**

|  |  |
| --- | --- |
| **Questions** | **Mark** |
| Part A Locally Install and Discover the System |  |
| 1. Identify current state of collateral | 5 |
| 2. Your Setup | 5 |
| 3. Questions i and ii | 2 |
| Part B -Initial Understanding of the System |  |
| 1. Users and roles | 5 |
| 2. Explain workflow for each role | 5 |
| 3. Issues | 5 |
| Part C - Deployment and Additional Setup |  |
| 1. Readme status/updates | 1 |
| 2. Azure project documentation | 1 |
| 3. Versioning | 2 |
| 4. Branching and merging | 2 |
| 5. Updating your Setup | 2 |
| 6. Deploy | 5 |
| Part D - Understanding the Backlog |  |
| 1. items to fix | 10 |
|  |  |
| Teammates assessment (given) | 4 |
| Teammates assessment (received) | 4 |
| Proper submission, English, Professional standards | 4 |
| **Total** | **62** |

1. Consider: Any updates required? Readme and other info completed? [↑](#footnote-ref-1)
2. Consider: What’s in the trunk vs the branches. Where is the current production/latest version published from? Which trunk/branch will you be working on? [↑](#footnote-ref-2)
3. Consider the condition of the Azure work items? Any stale or untrustworthy item (user stories, bugs, tasks)? Are all past milestones clean without lingering open items? [↑](#footnote-ref-3)