

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

|  |  |  |
| --- | --- | --- |
| **CMA Team** | | |
| **Project team** | Nguyen Thi Trang | SE05803 |
| Nguyen Duc Thien | SE05883 |
| Le Thi Thu Trang | SE05909 |
| Do Ngoc Khanh | SE06047 |
| Do Trung Duc | SE05844 |
| **Supervisor** | Mr. Bui Dinh Chien | |
| **Project code** | CMA | |

– Hanoi, August 2020 –

**Table of Contents**

[I. Project Report](#_heading=h.gjdgxs) **3**

[1. Status Report](#_heading=h.30j0zll) 3

[2. Team Involvements](#_heading=h.1fob9te) 3

[3. Issues/Suggestions](#_heading=h.3znysh7) 3

[II. Project Management Plan](#_heading=h.2et92p0) **4**

[1. Overview](#_heading=h.tyjcwt) 4

[1.1 WBS & Estimation](#_heading=h.3dy6vkm) 4

[1.2 Project Objectives](#_heading=h.1t3h5sf) 5

[1.3 Project Risks](#_heading=h.4d34og8) 6

[2. Management Approach](#_heading=h.2s8eyo1) 7

[2.1 Project Process](#_heading=h.17dp8vu) 7

[2.2 Quality Management](#_heading=h.3rdcrjn) 8

[2.3 Training Plan](#_heading=h.26in1rg) 9

[3. Master Schedule](#_heading=h.lnxbz9) 9

[4. Project Organization](#_heading=h.35nkun2) 10

[4.1 Team & Structures](#_heading=h.1ksv4uv) 10

[4.2 Roles & Responsibilities](#_heading=h.44sinio) 11

[5. Project Communication](#_heading=h.2jxsxqh) 14

[5.1 Communication Plan](#_heading=h.z337ya) 14

[5.2 External Interface](#_heading=h.3j2qqm3) 14

[a. FU Contacts](#_heading=h.b5w5nnuwret1) 14

[b. Customer Contacts](#_heading=h.oj5g71r7oa4h) 15

[6. Configuration Management](#_heading=h.1y810tw) 15

[6.1 Tools & Infrastructures](#_heading=h.4i7ojhp) 15

[6.2 Document Management](#_heading=h.2xcytpi) 16

[6.3 Source Code Management](#_heading=h.3whwml4) 17

# I. Project Report

## 1. Status Report

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Work Item** | **Status** | **Notes (Work Item in Details)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

## 2. Team Involvements

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

## 3. Issues/Suggestions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Issue** | **Status** | **Notes (Solution, Suggestion, etc.)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

# II. Project Management Plan

## 1. Overview

### 1.1 WBS & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| ***1*** | ***Existed systems research*** |  | ***10*** |
| 1.1 | Identify the theme of the Capstone project | Complex | *5* |
| 1.2 | Contact with end users of existed systems | Complex | *4* |
| 1.3 | Defines project name | Simple | *1* |
| ***2*** | ***Initiating*** |  | ***9*** |
| 2.1 | Kick-off | Simple | 1 |
| 2.2 | Identify stakeholder | Complex | 4 |
| 2.3 | Develop project introduction document | Medium | 4 |
| ***3*** | ***Planning*** |  | ***17*** |
| 3.1 | Define scope | Simple | 2 |
| 3.2 | Choose working model process | Medium | 1 |
| 3.3 | Choose tools & techniques | Complex | 1 |
| 3.4 | Develop project schedule | Medium | 1 |
| 3.5 | Development project resource | Medium | 3 |
| 3.6 | Define risk project | Medium | 4 |
| 3.7 | Develop project plan document | Medium | 5 |
| ***4*** | ***Analysis*** |  | ***26*** |
| 4.1 | Gather Requirements | Complex | 5 |
| 4.2 | Training Strategy | Medium | 4 |
| 4.3 | Implementation Strategy | Medium | 4 |
| 4.4 | Create Requirements Management Plan | Simple | 1 |
| 4.5 | Capture Additional Client Information | Medium | 4.5 |
| 4.6 | Build Conceptual Systems Design | Complex | 2.5 |
| 4.7 | Develop Software Requirement document | Medium | 5 |
| ***5*** | ***Design*** |  | ***31*** |
| 5.1 | Ensure Architectural Foundation | Medium | 5 |
| 5.2 | Validate Standards and Guidelines | Complex | 5 |
| 5.3 | Design the Solution | Medium | 6 |
| 5.4 | Build the Technical Systems Design | Complex | 5 |
| 5.5 | Review the Design | Medium | 5 |
| 5.6 | Develop Software Design Document | Medium | 5 |
| ***6*** | ***Implementation*** |  | ***40*** |
| 6.1 | Prepare for Implementation | Simple | 5 |
| 6.2 | Perform Training | Medium | 5 |
| 6.3 | Implement the Solution | Complex | 20 |
| 6.4 | Pilot Test the Solution | Medium | 5 |
| 6.5 | Monitor the Solution | Simple | 5 |
| ***7*** | ***Testing*** |  | ***24*** |
| 7.1 | Validate Test Coverage | Simple | 5 |
| 7.2 | Integration Testing | Complex | 5 |
| 7.3 | System Testing | Complex | 5 |
| 7.4 | User Acceptance Testing | Complex | 4 |
| 7.5 | Develop Software User Guides Document | Medium | 5 |
| ***8*** | ***Deployment*** |  | ***3*** |
| 8.1 | Deploy on server |  | 3 |
| ***Total Estimated Effort (man-days)*** | | | ***160*** |

### 1.2 Project Objectives

Provide the overall project objective description and then the specific target metrics of your project in terms of time, cost, and quality. For example

* Timeliness (%):
* Allocated Effort (man-days): 160
* Defect Distribution:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Quality Stage** | **No. of Defects** | **% of Defect** | **Notes** |
| 1 | Reviewing | 100 | 15 |  |
| 2 | Unit Test | 250 | 20 |  |
| 3 | Integration Test | 280 | 35 |  |
| 4 | System Test | 320 | 25 |  |
| 5 | User Acceptance Test | 50 | 5 |  |
| ***Total*** | | ***1000*** | ***100%*** |  |

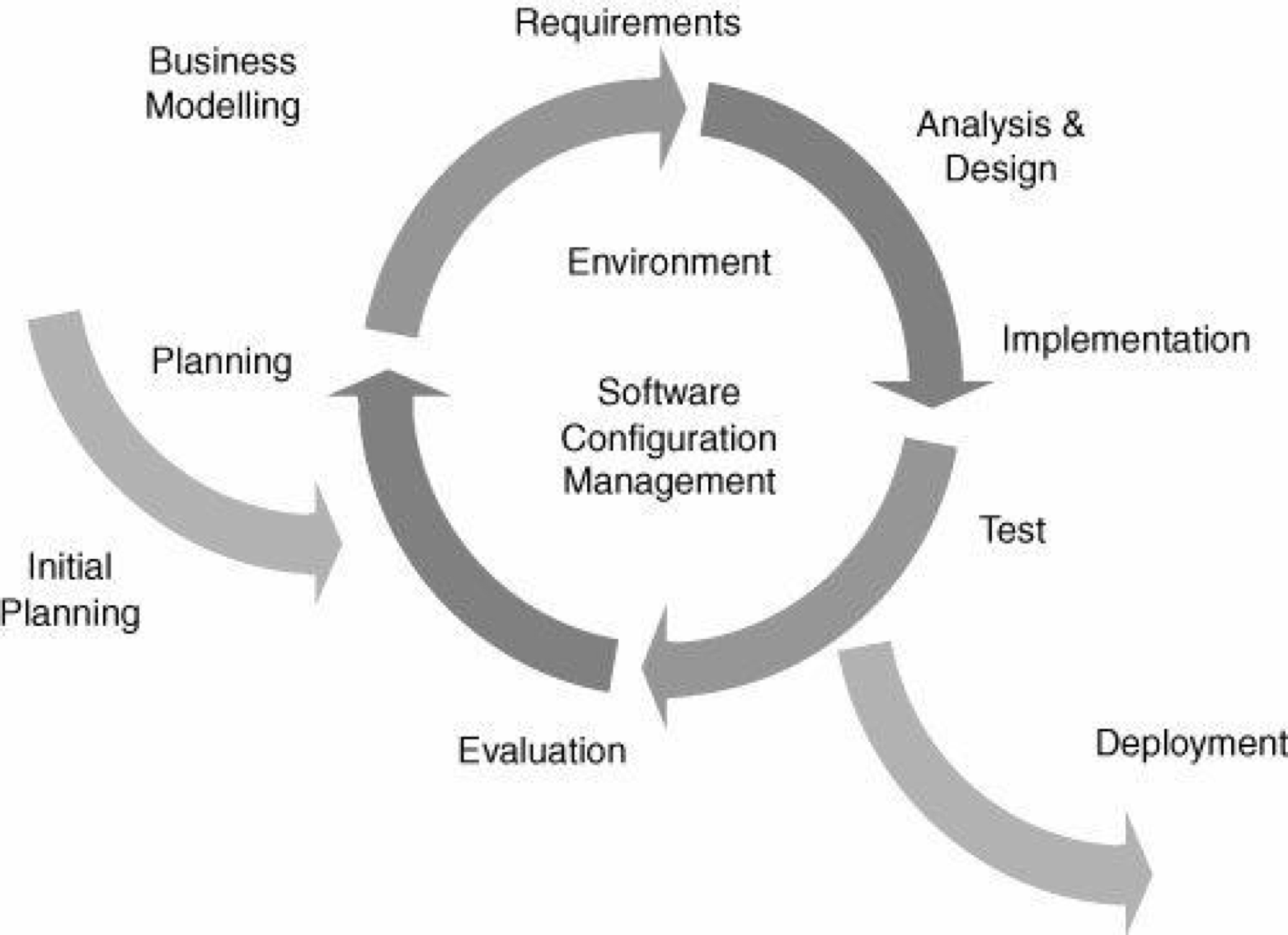
### 1.3 Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Project team member do not meet deadlines | High | Medium | Other members support if the task is an important task |
| 2 | Data loss | High | Medium | Restore backed up data from GitHub.  If a requirement has a new update, all members have to join the meeting to be aware and make decisions. |
| 3 | Conflict between team members | High | High | Sit down and talk frankly |
| 4 | Requirement  changed | Medium | Medium | Discuss with the Supervisor to decide which requirements should be implemented.  Change requirement. Develop for new required func |
| 5 | Spirit goes down | Low | High | Communicate regularly to make people more close |
| 6 | Lack of skill and knowledge for a specified work | Medium | High | Learn more courses on coursera |
| 7 | Team members may distraction | High | High | Understand team members' schedule and assign suitable tasks and set appropriate deadlines.  Require team members to set high priority for the project. |
| 8 | Team members do not understand about requirements | Medium | High | Every members when join develop the project must join the develop for SRS |
| 9 | Lack of Supervisor support | Medium | Low | Define a meeting schedule with the Supervisor.  Ask for support from other sources. |
| 10 | The designed database may be a failure. | High | High | Team members researchers more about how to design a database.  Team members discuss and review the project database.  Ask the support from the supervisor or the database expert |
| 11 | Business problem | Medium | High | Make sure the business logic of any ideas is carefully analyzed. |
| 12 | Source code may be conflicted | Low | High | Pull source code before committing the source code.  Use the backup version, discuss with other members and continue to work. |

## 2. Management Approach

### 2.1 Project Process

Clinic management application is develop according to the following model:



*Figure 1: Software Process Model*

The Iterative and Incremental Software Process Model is mostly used when the scope of the project is big, the major requirements are defined clearly, some more details will be added later in software development. By using this software process model, we break down the developing system task into a series of smaller tasks which will be completed separately, allowing us to take advantage of what was learned during development of earlier parts of the system. In addition, the iterative model is easier than other models when the issues are discovered. They are fed back to the team, and solutions will be found while the project is still in development.

### 2.2 Quality Management

Some of the quality approach :

* **Defect Prevention:** Discuss with the team before implementing a feature for exceptional cases that could occur, use pair programming to handle defects in the coding phase.
* **Reviewing:** Apply reviewing at different level:
  + Self review: Member reviews his/her source code by himself/herself
  + Peer review: Each member reviews other members’ source code
  + Final review: Leader review source code of his/her team and merge if there are no problems left
* **Unit testing:** Create unit test for api provided.
* **Integration testing:** is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.
* **System testing :** is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.

### 2.3 Training Plan

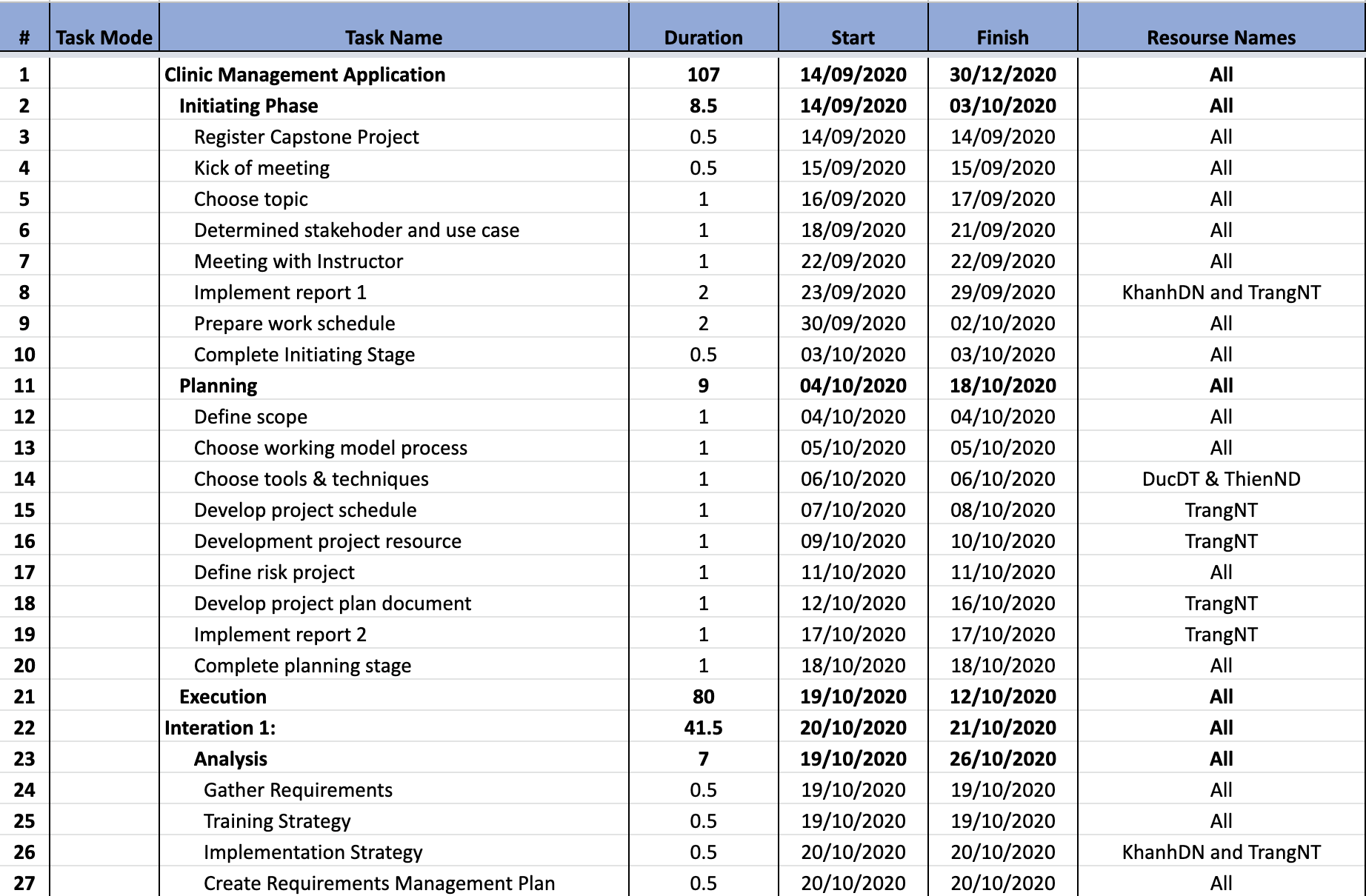
To be sure all members must have knowledge of code and members understand requirements of business. We have organized training for all members based on the plan below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| Git, Github | All | 09/28, 1 hour | Mandatory |
| PostgreSQL | All | 09/28, 1 hour | Mandatory |
| Java Spring Boot | All | 09/29, 3 hour | Mandatory |
| Angular | All | 09/30, 4 hour | Mandatory |

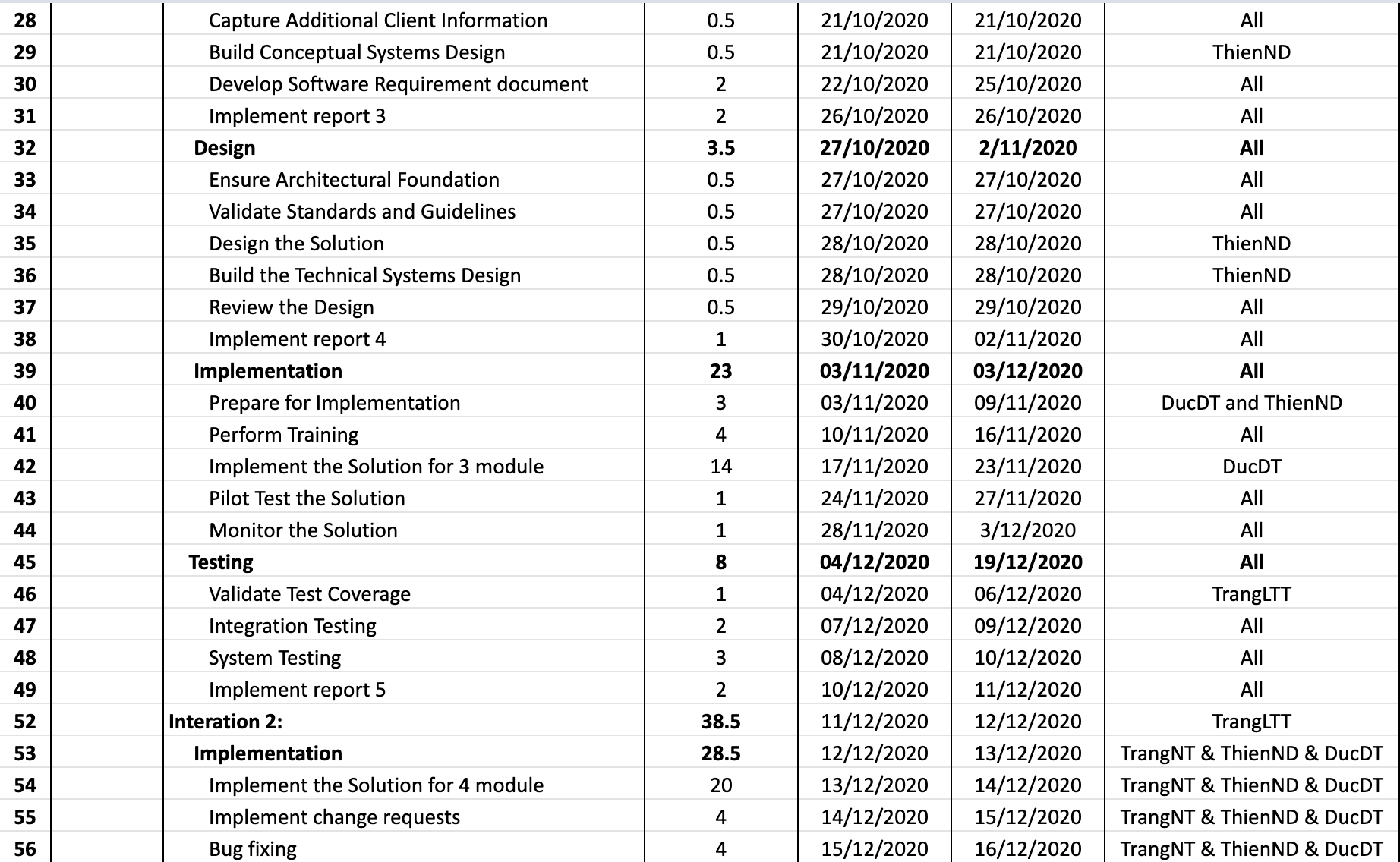
## 3. Master Schedule

Control objectives are checked according to the scheme's mode. We have posed a clear set of project’s schedule for completion deadlines.

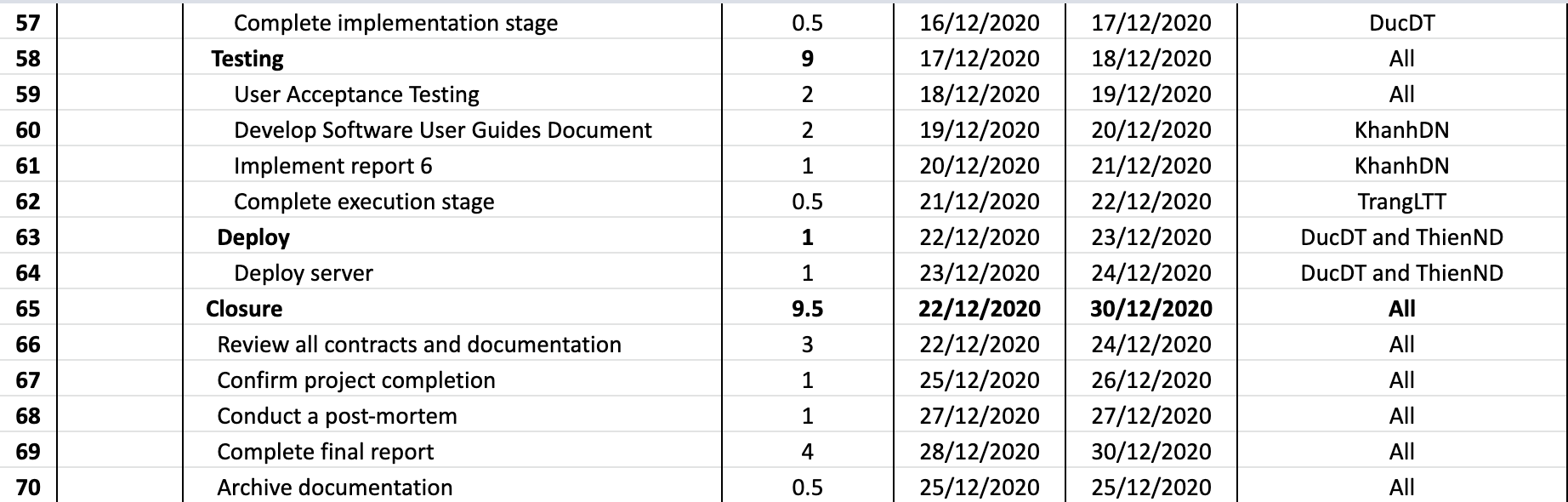
|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Deliverable** | **Due Date** | **Deliverable Scope** |
| 1 | Initiating Phase | 20 | The role of stakeholders.  Collect customer requests.  Define the objectives, scope and requirements of the project.  Project kick-off meeting. |
| 2 | Project Plan | 15 | Establish detailed WBS for the project.  Make a specific task list.  Scheduling. |
| 3 | SRS | 7 | Define Use cases, business rules, ERD . |
| 4 | Design | 7 | Architecture Design, Detailed Design, Database. |
| 5 | Implementation | 30 | Coding & Unit testing. |
| 6 | Testing | 18 | System testing, Integration Testing, Acceptance testing. |
| 7 | Deploy | 2 | Deploy on the server. |
| 8 | Closure | 10 | Solve project backlog.  Complete documentation project.  Handing over the project results to the end user. |



*Figure 2.1: CMA Schedule*



*Figure 2.2: CMA Schedule*

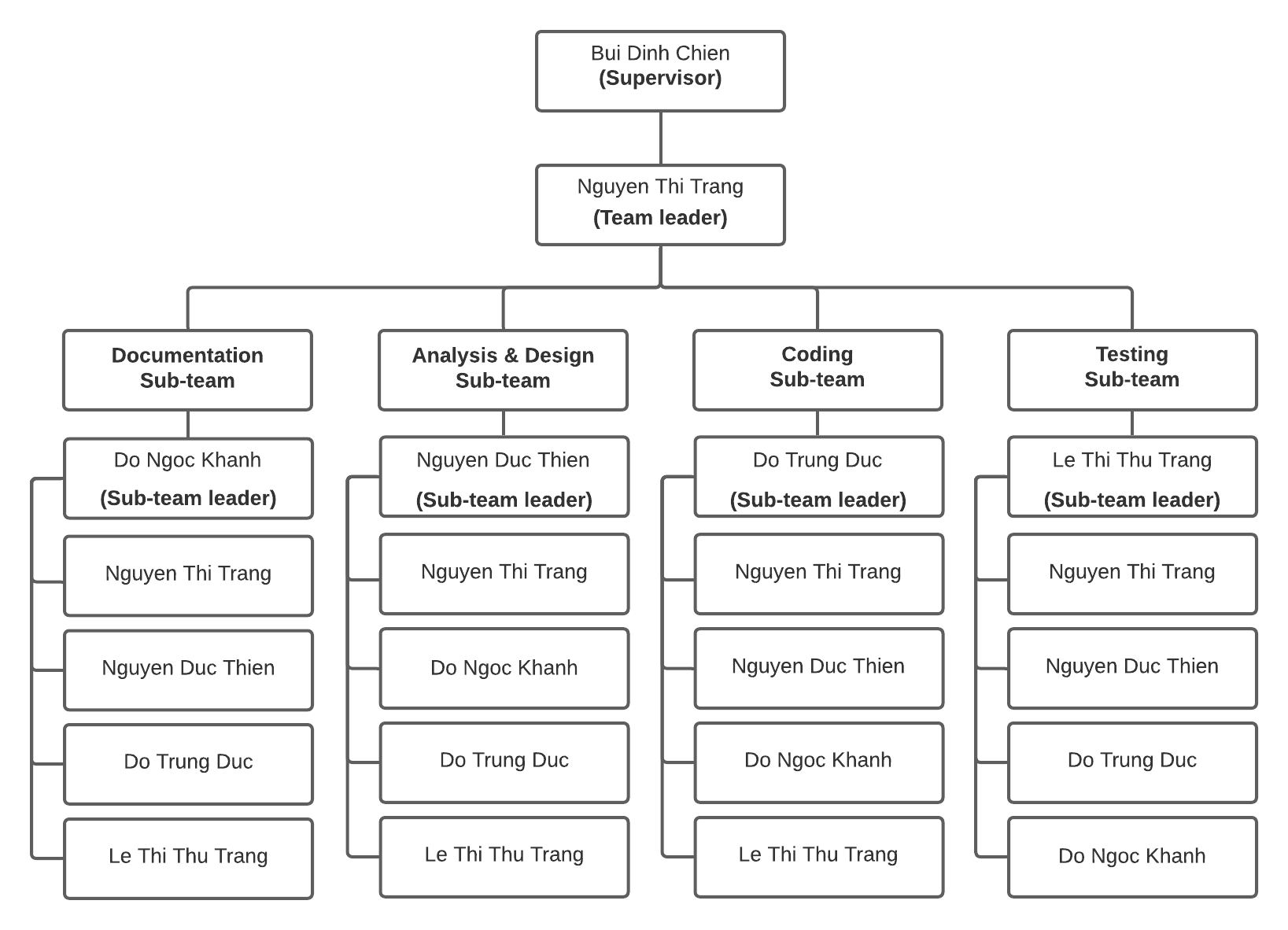


*Figure 2.3: CMA Schedule*

## 4. Project Organization

### 4.1 Team & Structures

CMATeam organization structure above includes 4 sub-teams: Documentation sub-team, Analysis & Design sub-team, Coding sub-team and Testing sub-team. Each sub-team has a sub-team leader and sub-team members:



*Figure 3: Team & Structures*

### 4.2 Roles & Responsibilities

This is the detail description about role and responsibility of each member in the organization structure:

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Responsibility** |
| Team leader | Nguyen Thi Trang | Have overall responsibility for the project and assign tasks to members.  Tracking member’s work.  Report working status to the instructor.  Managing reports and necessary documentation.  Controlling time management. |
| **Documentation Team** | | |
| Documentation Sub-team leader | Do Ngoc Khanh | Complete all of the documents.  Project Introduction.  Software Requirement Specification.  Software User Guides.  Presentation Slide. |
| Member #1 | Nguyen Thi Trang | Project Management Plan.  Presentation Slide. |
| Member #2 | Nguyen Duc Thien | Software Design Document. |
| Member #3 | Do Trung Duc | Help Software Requirement Specification. |
| Member #4 | Le Thi Thu Trang | Test plan document.  Installation guide. |
| **Analysis & Design Team** | | |
| Analysis & Design Sub-team leader | Nguyen Duc Thien | System Architecture design.  Screen design.  Sequence diagram.  Create use case specification. |
| Member #1 | Nguyen Thi Trang | Help create use case specifications.  Help Screen design.  Help database design.  Create a Class diagram. |
| Member #2 | Do Ngoc Khanh | Help ERD Diagram. |
| Member #3 | Do Trung Duc | ERD Diagram.  Database design.  Database Dictionary. |
| Member #4 | Le Thi Thu Trang | Use case specification.  Help Class diagram. |
| **Coding Team** | | |
| Coding Sub-team leader | Do Trung Duc | Decide technique and tools to use.  Train other members about web development.  Keeping track of development work done by other  coding team members.  Coding back-end & front-end.  Merge code.  Control source code. |
| Member #1 | Nguyen Thi Trang | Coding back-end. |
| Member #2 | Nguyen Duc Thien | Coding front-end.  Deploy web application to host server.  Help Decide technique and tools to be used. |
| Member #3 | Do Ngoc Khanh | Coding back-end. |
| Member #4 | Le Thi Thu Trang | Coding front-end. |
| **Testing team** | | |
| Testing Sub-team leader | Le Thi Thu Trang | Create a test plan.  Responsible for the test plan.  Controlling testing activities.  Create test reports.  Manage test resources and assign test tasks Create unit test and bug log.  Create test cases (Acceptance test).  Report test result.  Implement test cases. |
| Member #1 | Nguyen Thi Trang | Help create test case (Integration test, System test,  Acceptance test). |
| Member #2 | Nguyen Duc Thien | Help create test case (Integration test, System test,  Acceptance test).  Help implement test cases. |
| Member #3 | Do Trung Duc | Help create test cases (Acceptance test).  Help implement test cases. |
| Member #4 | Le Thi Thu Trang | Help create test cases (Integration test, System test, Acceptance test). |

## 5. Project Communication

### 5.1 Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| Weekly meeting schedule | All member and Supervisor | Assign tasks.  Solve the problems.  Review report. | Slot 2 wednesday, once per week | Offline |
| Daily meeting schedule | All member | Check the progress of groups | 20h, everyday | Google meet |
| Unscheduled meeting | All member | Handle important work |  | Google meet, offline. |
| Training | All member | New training of knowledge for members | Slot 2 & 3, monday and wednesday and friday. | Offline, google meet. |
| Communication channel | All member | Assign tasks in the team  Means of communication, work management. | Everyday | Trello, facebook group, phone, google meet. |

### 5.2 External Interface

#### a. FU Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person**  **(name, position)** | **Contact address**  **(email, telephone)** | **Responsibility** |
| Supervisor | Bui Dinh Chien | ChienBD@fe.edu.vn  sdt….. | Provide document template  Give instruction to project team  Review deliverables  Supervise project status |
| Supporter | Ngo Thi Vinh Ha | HaNTV2@fe.edu.vn | Provide document template  Receive report project  Answer questions about the project |

#### b. Customer Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person**  **(name, position)** | **Contact address**  **(email, telephone)** | **Responsibility** |
| Supporter | Nguyễn Trần Kiên | 096.159.1593 | Support for the research system of business and analysis exist. |
| Customer | Nguyễn Thị Hạnh | 091.648.3289 | Acceptance of products and supply requirements. |

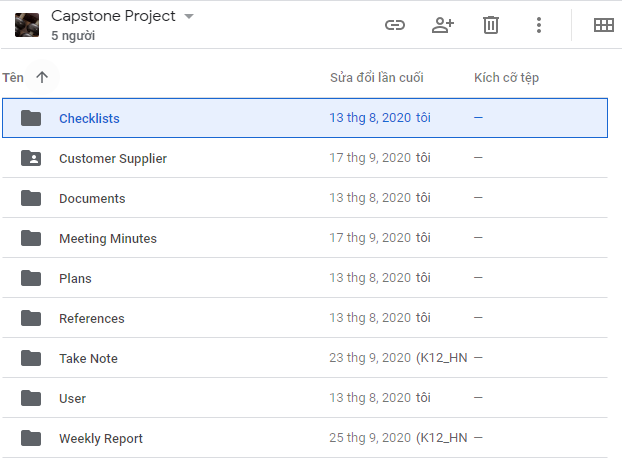
## 6. Configuration Management

### 6.1 Tools & Infrastructures

|  |  |
| --- | --- |
| **Programming languages** | Java, Typescript |
| **Framework** | Java Spring Boot, Angular |
| **DBMS** | PostgreSQL |
| **IDEs/Editors** | Visual Studio Code, Eclipse |
| **UML tools** | Lucidchart.com |
| **Version Control** | Github, Google drive |
| **Deployment server** | Google compute engine, Docker |
| **Project management tool** | Trello.com |
| **Mockup design tool** | Draw.io, figma.com |

### 6.2 Document Management

Document will be managed through Google Drive. We have created shared memory that has the following directory structure:



* **Checklists**: Store checklist files (coding review, test review, coding review, ...)
* **Customer Supplier**: Contains files, documents, images provided by customer
* **Documents**: Contains all document files of the project (reports, design documents, api documents, test documents, ...)
* **Meeting Minutes**: containing Meeting Minutes report files weekly
* **Plans:** Contains document files related to project planning
* **References**: Contains reference document files
* **Take Note**: Take note files after each team meeting
* **User**: Include subfolders for each member. Each folder will contain separate documents or work in progress
* **Weekly Report**: Containing Weekly Report document

### 6.3 Source Code Management

Source code will be managed through Github

* We have two main branches with an infinite lifetime:
  + Main: to be the main branch where the source code of HEAD always reflects a production-ready state.
  + Develop: to be the main branch where the source code of HEAD always reflects a state with the latest delivered development changes for the next release
* Each Feature will be deployed to work on a branch  
  Feature branch are used to develop new features for the upcoming or a distant future release  
  May branch off from: develop  
  Must merge back into: develop  
  Branch naming convention: anything except master, develop, \*-feature
* Hotfix branches are very much like release branches in that they are also meant to prepare for a new production release, albeit unplanned  
  May branch off from: master  
  Must merge back into: develop and master  
  Branch naming convention: \*-hotfix