

# PORTFOLIO

## Assignments for the course: Project: Software Engineering (DLMCSPSE01)

### TABLE OF CONTENTS

<b>1. TOPICS AND TASKS .....</b>	<b>2</b>
<b>1.1. Task 1: Design and development of a web application .....</b>	<b>2</b>
1.1.1. Conception phase.....	3
1.1.2. Development phase/reflection phase .....	3
1.1.3. Finalization phase .....	4
<b>1.2. Task 2: Design and development of a mobile application.....</b>	<b>5</b>
1.2.1. Conception phase.....	6
1.2.2. Development phase/reflection phase .....	7
1.2.3. Finalization phase .....	7
<b>2. TUTORIAL SUPPORT .....</b>	<b>8</b>
<b>3. EVALUATION .....</b>	<b>9</b>
<b>4. FORMAL GUIDELINES AND SPECIFICATIONS FOR SUBMISSION .....</b>	<b>10</b>
4.1. Components of the examination performance .....	10
4.2. Format for Digital File Submission .....	11
4.3. Format of Abstract .....	13

## 1. TOPICS AND TASKS

Within the framework of this course, one of the following topics must be selected.

### 1.1. Task 1: Design and development of a web application

Digital transformation is changing the way how we work and live, how we do business. Software enables new business models, improves business processes, introduces digitized products and services. Nowadays, ready-to-use frameworks and technologies allow software engineers to quickly develop and deploy new, innovative applications. In this project, you will design and develop such a web application.

It is up to you and your creativity to determine what type of application you will develop. However, your project and the upcoming application must satisfy the following criteria:

- Your application must **deliver customer value**, i.e., it must have a clear and significant benefit to a defined target group.
- Your upcoming solution must be something **new**. You are not allowed to simply copy an existing application.
- You must implement a reasonable amount of **code**. A mock-up with slides or a simple configuration of an existing software is not sufficient as final product. Such a step can be helpful in the conception phase to explore and discuss alternatives.
- The application must **run in a web browser**.

In this course, **you will work through all stages of a software development project**. It is important that you select the appropriate methodologies, techniques, and tools at every stage and for every task. This means that you will apply the knowledge in the fields of requirements specification, software design, implementation, and testing that you have acquired in prior modules.

As a considerable percentage of software engineering projects fails to deliver in time, quality, and budget, you should take appropriate actions to minimize risks in every phase.

The following **acceptance criteria** must be met:

- The tools, programming language and libraries used to implement your application are your own choice. However, you are not permitted to copy or modify existing, third-party applications that you might find on the internet or elsewhere. You can use third-party libraries if you indicate them correctly.
- The application must come with a web-based graphical user interface that is self-explanatory and that can be intuitively used by potential customers.
- The solution delivered must be valuable to your customers. It must provide the most important features and solve the most basic problems of your target-group. You must provide enough and sensible (test) data (no dummy text) so that the application can directly be used, and its benefits become tangible.
- The application must meet the non-functional requirements. If this is not possible in the scope of the project, it must be described as technical debt.
- The software must come up with an appropriate project documentation, including a project profile, requirements documents, a documentation of the software and system architecture.
- Version and release control of the software and the corresponding architecture documentation must be established.
- The source code and all necessary files must be provided in GitHub. Hosting the project on GitHub is part of building your portfolio.

- Web applications must be provided as cloud-hosted application (e.g. AWS, Google Cloud, ...) so that the front end can directly be accessed.
- The software must come with installation and run instructions.

Your application needs to be designed, built, documented, and delivered according to the **following three phases**.

### 1.1.1. Conception phase

This phase is very important for the success of your project. Anything that is overlooked or forgotten in this phase has a negative effect on the implementation later and will lead, in the worst case, to useless results.

**The first step is to create a written "project profile"** for your software engineering project. It includes the objectives and the scope of the project, your target group, the risks associated with the project, a project plan, and the project organization. In companies, such a project profile is typically required as input for the decision on its funding. Due to the scope of this course, it is not necessary to calculate a complete business case.

During the conception phase, you must decide on the **software development methodology** that you will follow in the development phase. Briefly explain your choice. Note that depending on the chosen methodology, stages and roles in the project will vary.

In the next step, you will analyze the **functional and non-functional requirements** for your upcoming application. Ensure an adequate level of detail corresponding to the selected software development methodology. In a **glossary**, you will define the most important terms to foster a common understanding between all stakeholders involved in your project or its review. Think about how you will test your software regarding these requirements.

On this basis, you start the **system design**. You show the scope and the context of the system (e.g., in UML) and decide about technologies and tools. Explain your motivation - why you decided to choose this way? Moreover, you will create at least one diagram (e.g., in UML) that shows the decomposition of the overall system into its building blocks (such as modules, components, frameworks, layers, etc. ...) and the dependencies between these building blocks. This view is a very important part of an architecture documentation. In analogy to the architecture of a house, this view corresponds to the floor plan.

As submission, you prepare a written (approx. 1/2 DIN A4 page) **summary of the project profile and your considerations with respect to the chosen software development methodology**. Additionally, you provide all other artifacts that you created as described above as a **composite presentation PDF**. This file should contain visual elements that facilitate comprehension and present information in a structured way.

Throughout the process, online tutorials are offered, and they provide an opportunity to talk, share ideas and/or drafts, and obtain feedback. In the online tutorials, exemplary work can be discussed with the tutor. Here, everyone has the opportunity to get involved and learn from each other's feedback. **It is recommended to make use of these channels to avoid errors and to make improvements.** You should only submit work after making use of the above-mentioned tutorial and informative media. This will be followed by a feedback from the tutor and the work on the second phase can begin.

### 1.1.2. Development phase/reflection phase

In this phase, **the implementation starts**. In addition, there are several activities to ensure that the developed application meets customer requirements, that it can be deployed, maintained, and further developed. Take care

that you establish **version and release control** of the software and the corresponding architecture documentation.

- You set up the frameworks and tools that you described in the conception phase.
- You select third-party libraries you can build upon.
- You refine the requirements gathered in the conception phase.
- You implement the modules/components outlined in your building block diagram.
- You further develop your software and system documentation.
- You document important design decisions including rationales.
- You comment your source code to make it easier for humans to understand.
- You prepare test data and perform adequate software tests.

In this phase you must submit an **explanation of your design and implementation procedure** in written form (approx. 1/2 page). Additionally, you provide a **composite presentation PDF** including the updated requirements, the complemented architecture documentation, and references and hyperlinks to the frameworks, third-party libraries, and tools you decided to use as well as the rationale behind your decision. The file should contain visual elements that facilitate comprehension and present information in a structured way.

Throughout the process, online tutorials and other channels provide the opportunity to profoundly discuss ideas and/or drafts and to get sufficient feedback, tips, and hints. **It is recommended to use these channels to avoid errors and to improve your work.** Once this is done, you can hand in your second phase for evaluation. Following a feedback from the tutor, your work on the final draft will continue in the third phase.

### 1.1.3. Finalization phase

In this final phase, you prepare everything for the final submission. Your goal is to polish and refine your application and its documentation. Certain elements may have to be improved or changed to finalize the task and complete this portfolio course. It is important that you consider the feedback that you received on your submission at the end of the previous phase. Depending on the changes you make in your software, it might be necessary to provide an updated version of the documentation of the software and system architecture. In addition - if applicable - you provide a list of the technical debts. Make sure that your software comes with installation and run instructions so that it is clear how to install the application and how to use your software. In addition, your application must satisfy the following criteria:

- Web applications must be provided as docker-compose configuration (.yml) including all relevant files.
- Web applications must be provided as cloud-hosted application (e.g., AWS, Google Cloud, ...) so that the front end can directly be accessed. If you restrict the access, make sure that you include the login credentials in the submitted documentation.

Your finished application is submitted by providing all files that you created, your program code, documentation installation manual etc., as follows:

- Your project is hosted on a public GitHub repository. That means you must create a GitHub account, if you don't have one yet, and create a repository.
- You upload all your code and all necessary files into this repository.
- On submission in PebblePad you provide a link to the repository.
- You create a ZIP file from all files contained in the GitHub repository and put it into a folder. You must zip this folder and paste it into your submission in Pebble Pad.

**Important note:** The content of the GitHub repository and the contents of the ZIP file uploaded to your zip folder should be identical. You are not allowed to modify either one after the final submission. The content of the zip folder will be sent to the examination office upon submission.

Now, it is time to look back at your software engineering project and do a “**lessons learned**”. You should especially reflect on the applied tools, methodologies, and techniques and on your handling of the resources.

For the submission, you provide a **2-page abstract PDF document** in which you describe your solution in terms of content and concept. This abstract presents a short break-down (“making of” of the project) about the technical approach in a clear and informative way. It also includes the main insights from your lessons learned. Moreover, you must prepare a **.txt file in which you include the link to your public GitHub repository**. In the case of a web application, you must also include the link to the application and if applicable the login data. In addition, you provide all project documentation including the ZIP file with all files contained in the GitHub repository in your zip folder. The structure of this folder is defined in chapter 4.2.

In the “Finalization phase”, the online tutorials and other channels also provide the opportunity to obtain sufficient feedback, tips, and hints before the finished product is finally handed in. **It is recommended to use these channels to avoid errors and to make improvements.** The **finished product** is submitted with the results from Phase 1 and Phase 2 and together with the materials mentioned above. Following the submission of the third portfolio page, the tutor submits the final feedback which includes evaluation and scoring within six weeks.

## 1.2. Task 2: Design and development of a mobile application

Digital transformation is changing the way how we work and live, how we do business. Software enables new business models, improves business processes, introduces digitized products and services. Nowadays, ready-to-use frameworks and technologies allow software engineers to quickly develop and deploy new, innovative applications. In this project, you will design and develop such a mobile application.

It is up to you and your creativity to determine what type of application you will develop. However, your project and the upcoming application must satisfy the following criteria:

- Your application must **deliver customer value**, i.e., it must have a clear and significant benefit to a defined target group.
- Your upcoming solution must be something **new**. You are not allowed to simply copy an existing application.
- You must implement a reasonable amount of **code**. A mock-up with slides or a simple configuration of an existing software is not sufficient as final product. Such a step can be helpful in the conception phase to explore and discuss alternatives.
- The application must **run on a mobile device** (smartphone, tablet).

In this course, **you will work through all stages of a software development project**. It is important that you select the appropriate methodologies, techniques, and tools at every stage and for every task. This means that you will apply the knowledge in the fields of requirements specification, software design, implementation, and testing that you have acquired in prior modules.

As a considerable percentage of software engineering projects fails to deliver in time, quality, and budget, you should take appropriate actions to minimize risks in every phase.

The following **acceptance criteria** must be met:

- The tools, programming language and libraries used to implement your application are your own choice. However, you are not permitted to copy or modify existing, third-party applications that you might find on the internet or elsewhere. You can use third-party libraries if you indicate them correctly.
- The application must come with a graphical user interface (mobile) that is self-explanatory and that can be intuitively used by potential customers.
- The solution delivered must be valuable to your customers. It must provide the most important features and solve the most basic problems of your target-group. You must provide enough and sensible (test) data (no dummy text) so that the application can directly be used, and its benefits become tangible.
- The application must meet the non-functional requirements. If this is not possible in the scope of the project, it must be described as technical debt.
- The software must come up with an appropriate project documentation, including a project profile, requirements documents, a documentation of the software and system architecture.
- Version and release control of the software and the corresponding architecture documentation must be established.
- The source code and all necessary files must be provided in GitHub. Hosting the project on GitHub is part of building your portfolio.
- Mobile applications can be implemented as native or hybrid apps.
- Mobile applications must be provided as .apk file for android applications or .ipa file for iOS.
- The software must come with installation and run instructions.

Your application needs to be designed, built, documented, and delivered according to the **following three phases**.

### 1.2.1. Conception phase

This phase is very important for the success of your project. Anything that is overlooked or forgotten in this phase has a negative effect on the implementation later and will lead, in the worst case, to useless results.

**The first step is to create a written "project profile"** for your software engineering project. It includes the objectives and the scope of the project, your target group, the risks associated with the project, a project plan, and the project organization. In companies, such a project profile is typically required as input for the decision on its funding. Due to the scope of this course, it is not necessary to calculate a complete business case.

During the conception phase, you must decide on the **software development methodology** that you will follow in the development phase. Briefly explain your choice. Note that depending on the chosen methodology, stages and roles in the project will vary.

In the next step, you will analyze the **functional and non-functional requirements** for your upcoming application. Ensure an adequate level of detail corresponding to the selected software development methodology. In a **glossary**, you will define the most important terms to foster a common understanding between all stakeholders involved in your project or its review. Think about how you will test your software regarding these requirements.

On this basis, you start the **system design**. You show the scope and the context of the system (e.g., in UML) and decide about technologies and tools. Explain your motivation - why you decided to choose this way? Moreover, you will create at least one diagram (e.g., in UML) that shows the decomposition of the overall system into its building blocks (such as modules, components, frameworks, layers, etc. ...) and the dependencies between these building blocks. This view is a very important part of an architecture documentation. In analogy to the architecture of a house, this view corresponds to the floor plan.

As submission, you prepare a written (approx. 1/2 DIN A4 page) **summary of the project profile and your considerations with respect to the chosen software development methodology**. Additionally, you provide all other artifacts that you created as described above as a **composite presentation PDF**. This file should contain visual elements that facilitate comprehension and present information in a structured way.

Throughout the process, online tutorials are offered, and they provide an opportunity to talk, share ideas and/or drafts, and obtain feedback. In the online tutorials, exemplary work can be discussed with the tutor. Here, everyone has the opportunity to get involved and learn from each other's feedback. **It is recommended to make use of these channels to avoid errors and to make improvements.** You should only submit work after making use of the above-mentioned tutorial and informative media. This will be followed by a feedback from the tutor and the work on the second phase can begin.

### 1.2.2. Development phase/reflection phase

In this phase, **the implementation starts**. In addition, there are several activities to ensure that the developed application meets customer requirements, that it can be deployed, maintained, and further developed. Take care that you establish **version and release control** of the software and the corresponding architecture documentation.

- You set up the frameworks and tools that you described in the conception phase.
- You select third-party libraries you can build upon.
- You refine the requirements gathered in the conception phase.
- You implement the modules/components outlined in your building block diagram.
- You further develop your software and system documentation.
- You document important design decisions including rationales.
- You comment your source code to make it easier for humans to understand.
- You prepare test data and perform adequate software tests.

In this phase you must submit an **explanation of your design and implementation procedure** in written form (approx. 1/2 page). Additionally, you provide a **composite presentation PDF** including the updated requirements, the complemented architecture documentation, and references and hyperlinks to the frameworks, third-party libraries, and tools you decided to use as well as the rationale behind your decision. The file should contain visual elements that facilitate comprehension and present information in a structured way.

Throughout the process, online tutorials and other channels provide the opportunity to profoundly discuss ideas and/or drafts and to get sufficient feedback, tips, and hints. **It is recommended to use these channels to avoid errors and to improve your work.** Once this is done, you can hand in your second phase for evaluation. Following a feedback from the tutor, your work on the final draft will continue in the third phase.

### 1.2.3. Finalization phase

In this final phase, you prepare everything for the final submission. Your goal is to polish and refine your application and its documentation. Certain elements may have to be improved or changed to finalize the task and complete this portfolio course. It is important that you consider the feedback that you received on your submission at the end of the previous phase. Depending on the changes you make in your software, it might be necessary to provide an updated version of the documentation of the software and system architecture. In addition - if applicable - you provide a list of the technical debts. Make sure that your software comes with installation and run

instructions so that it is clear how to install the application and how to use your software. In addition, your mobile applications must be provided as **.apk file for android applications or .ipa file for iOS**.

Your finished application is submitted by providing all files that you created, your program code, documentation installation manual etc., as follows:

- Your project is hosted on a public GitHub repository. That means you must create a GitHub account, if you don't have one yet, and create a repository.
- You upload all your code and all necessary files into this repository.
- On submission in PebblePad you provide a link to the repository.
- You create a ZIP file from all files contained in the GitHub repository and put it into a folder. You must zip this folder and paste it into your submission in Pebble Pad.

**Important note:** The content of the GitHub repository and the contents of the ZIP file uploaded to your zip folder should be identical. You are not allowed to modify either one after the final submission. The content of the zip folder will be sent to the examination office upon submission.

Now, it is time to look back at your software engineering project and do a **“lessons learned”**. You should especially reflect on the applied tools, methodologies, and techniques and on your handling of the resources.

For the submission, you provide a **2-page abstract PDF document** in which you describe your solution in terms of content and concept. This abstract presents a short break-down (“making of” of the project) about the technical approach in a clear and informative way. It also includes the main insights from your lessons learned. Moreover, you must prepare a **.txt file in which you include the link to your public GitHub repository**. In addition, you provide all project documentation including the ZIP file with all files contained in the GitHub repository in your zip folder. The structure of this folder is defined in chapter 4.2.

In the “Finalization phase”, the online tutorials and other channels also provide the opportunity to obtain sufficient feedback, tips, and hints before the finished product is finally handed in. **It is recommended to use these channels to avoid errors and to make improvements.** The finished product is submitted with the results from Phase 1 and Phase 2 and together with the materials mentioned above. Following the submission of the third portfolio page, the tutor submits the final feedback which includes evaluation and scoring within six weeks.

## 2. TUTORIAL SUPPORT

In principle, several channels are open to attain feedback for the portfolios. The respective use is the sole responsibility of the user. The independent development of a product and the work on the respective portfolio parts is part of the examination performance and is included in the overall assessment.

On the one hand, the tutorial support provides feedback loops on the portfolio parts to be submitted in the context of the conception phase as well as the development and reflection phase. The feedback takes place within the framework of a submission of the respective part of the portfolio. In addition, regular online tutorials are offered. These provide you with an opportunity to ask any questions regarding the processing of the portfolio and to discuss other issues with the tutor. The tutor is also available for technical consultations as well as for formal and general questions regarding the procedure for portfolio management.

Technical questions regarding the use of “PebblePad” should be directed to the exam office via mail.



### 3. EVALUATION

The following criteria are used to evaluate the portfolio with the percentage indicated in each case:

Evaluation criteria	Explanation	Weighting
Problem Solving Techniques	*Capturing the problem *Clear problem definition/objective *Understandable concept	10%
Methodology/Ideas/Procedure	*Appropriate transfer of theories/models *Clear information about the chosen Methodology/Idea/Procedure	20%
Quality of implementation	*Quality of implementation and documentation	40%
Creativity/Correctness	*Creativity of the solution approach *Solution implemented fulfils intended objective	20%
Formal requirements	* Compliance with formal requirements	10%

The design and construction of the portfolio should take into account the above evaluation criteria, including the following explanations:

**Problem Solving Techniques:** Clear scope and goals as well as comprehensibility of the project, clear beneficiaries of the solution, use of appropriate technologies, libraries and services, understandable setup and execution of the application.

**Methodology/Idea/Procedure:** Appropriate selection and application of methods and techniques, consistent adherence to the chosen software development methodology, traceable use of methodologies, techniques and best practices, recognizable significant design and technology decisions, meaningful choices.

**Quality of implementation:** Visible concept of software and system architecture, implementation and testing, traceability of the vision, requirements and implemented features, comprehensibility of the architecture documentation and software, intuitive operation of the graphical user interface, compliance with the acceptance criteria outlined in chapter 3.

**Creativity/Correctness:** Addressing a previously unsolved customer problem with the project, new benefits and /or innovative features, correct understanding and implementation of the customer requirements, ensuring correct software by means of testing.

**Formal requirements:** The submission follows the acceptance criteria from chapter 3 and the formal guidelines provided in the “Guidelines for the creation of a portfolio”. It is particularly important to respect the formal submission requirements outlined in Chapter 4.

## 4. FORMAL GUIDELINES AND SPECIFICATIONS FOR SUBMISSION

### 4.1. Components of the examination performance

The following is an overview of the examination performance portfolio with its individual phases, individual performances to be submitted, and feedback stages at one glance. A template in “PebblePad” is provided for the development of the portfolio parts within the scope of the examination performance. The presentation is part of this examination.

Stage	Intermediate result	Performance to be submitted
Conception phase	Portfolio part 1	<ul style="list-style-type: none"> <li>• Concept presentation in written form (approx. 1/2 page), consisting of a summary of the project proposal and your considerations with respect to the chosen software development methodology</li> <li>• A composite presentation PDF including <ul style="list-style-type: none"> <li>○ the complete project profile,</li> <li>○ the requirements</li> <li>○ a glossary</li> <li>○ the system design with at least the system scope and context and a diagram with the building block view</li> </ul> </li> </ul>
Feedback		
Development phase/ reflection phase	Portfolio part 2	<ul style="list-style-type: none"> <li>• Explanation of your design and implementation procedure in written form (approx. 1/2 page)</li> <li>• A composite presentation PDF including <ul style="list-style-type: none"> <li>○ the updated requirements</li> <li>○ the complemented architecture documentation</li> <li>○ references and hyperlinks to the frameworks, third-party libraries, and tools as well as the rationale behind your choice</li> </ul> </li> </ul>
Feedback		
Finalization phase	Portfolio part 3	<ul style="list-style-type: none"> <li>• 2-page abstract (making of)</li> <li>• A .txt file including the link to your public GitHub repository, and – in the case of a web application – the link to the application and if applicable the login data</li> <li>• Upload the zip-folder (incl. all files and the final product with its installation manual according to the structure and content defined in chapter 4.2)</li> <li>• Result from phase 1</li> <li>• Result from phase 2</li> </ul>
Feedback + Grade		

## 4.2. Format for Digital File Submission

### Conception phase

Recommended tools/software for processing	<ul style="list-style-type: none"> <li>• Word, Powerpoint, Excel or LaTeX</li> <li>• Tools to draw software models (e.g. Visio, Enterprise Architect, SmartDraw, Edraw)</li> </ul>
Permitted file formats	PDF
File size	Max. 15 MB
Further formalities and parameters	<p>Files must always be named according to the following pattern:</p> <p><b>For the performance-relevant submissions on “PebblePad”:</b>  Name-First_Name_MatrNo_Course _P(hase)-1_S(ubmission)  Example: Mustermann-Max_12345678_ PSE _P1_S</p>

### Development/reflection phase

Recommended tools/software for processing	<ul style="list-style-type: none"> <li>• Word, Powerpoint, Excel or LaTeX</li> <li>• Tools to draw software models (e.g., Visio, Enterprise Architect, SmartDraw, Edraw)</li> <li>• Researched tool selection for task implementation</li> </ul>
Permitted file formats	PDF
File size	Max. 15 MB
Further formalities and parameters	<p>Files must always be named according to the following pattern:</p> <p><b>For the performance-relevant submissions on “PebblePad”:</b>  Name-First_Name_MatrNo_Course _P(hase)-2_S(ubmission)  Example: Mustermann-Max_12345678_ PSE _P2_S</p>

### Finalization phase

Recommended tools/software for processing	<ul style="list-style-type: none"> <li>• Word, Powerpoint, Excel or LaTeX</li> <li>• Tools to draw software models (e.g., Visio, Enterprise Architect, SmartDraw, Edraw)</li> <li>• Researched tool selection for task implementation</li> </ul>
Permitted file formats	<ul style="list-style-type: none"> <li>• PebblePad: PDF, TXT</li> <li>• GitHub: various file formats for your application and its documentation; PDF for other related documentation such as installation instructions, etc.</li> <li>• Zip folder: various file formats for submissions from all three phases. The code submitted via GitHub also must be added to the zip folder by creating a ZIP file from all the files.</li> </ul>
File size	as small as possible
Further formalities and parameters	The practical work must be submitted on GitHub.

**IMPORTANT** is the upload of the zip folder that has been created especially for the submission (please follow the instructions on myCampus). This folder contains all the files you used to complete the task. To ensure a better overview, please create subdirectories for this purpose.

#### The folder structure then looks like this:

- Main directory (name of the zip folder) -> *Name: Name-First\_Name\_Matriculation\_Course*
  - Subdirectory (these are the project profile, project plans and further documents related to the planning and the management of the project) -> *Name: 01-Project-management*
  - Subdirectory (these are requirements documents, glossary etc.) -> *Name: 02-Requirements*
  - Subdirectory (the software and system documentation, all diagrams etc.) -> *Name: 03-Architecture-documentation*
  - Subdirectory (the program code and related documentation, corresponding to your upload in the GitHub repository) -> *Name: 04-Implementation*. In this subdirectory, you upload a zipped version (ZIP file with all program code) with the following naming convention: *Name-First-Name\_MatrNo\_Course\_Submission\_Code.zip*

Please make sure that you either embed the images (and fonts, if any) linked in your document or to place them in the respective directory. Otherwise your documents cannot be opened completely and therefore cannot be assessed!

For the submission of your code on GitHub you are free to choose your own username. You create a public GitHub repository of your choice and upload all your code to it. That means your code should be available under the following URL schema:

[https://github.com/<user\\_name>/<repository\\_name>](https://github.com/<user_name>/<repository_name>)

Example: [https://github.com/maxmustermann/pse\\_myproject](https://github.com/maxmustermann/pse_myproject)

Files must always be named according to the following pattern:

#### For the performance-relevant submissions on “PebblePad”:

Name-FirstName\_MatrNo\_Course\_P(hase)-3\_S(submission)

Example: Mustermann-Max\_12345678\_PSE\_P3\_S

#### **4.3. Format of Abstract**

Length	2 pages of text
Paper size	DIN A4
Margins	Top and bottom 2cm; left 2cm; right 2cm
Font	General Text - Arial 11 pt.; Headings - 12 pt., Justify
Line Spacing	1,5
Sentences	Justified; hyphenation
Footnotes	Arial 10 pt., Justify
Paragraphs	According to mental structure - 6 pt. after line break
Affidavit	The affidavit shall be made in electronic form via “myCampus”. No submission of the examination performance is possible before it.

Please follow the instructions for submitting a portfolio on “myCampus”.

If you have any questions regarding the submission of the portfolio, please contact the exam office via mail.

Please also note the instructions for using PebblePad & Atlas!

**Good luck creating your portfolio!**