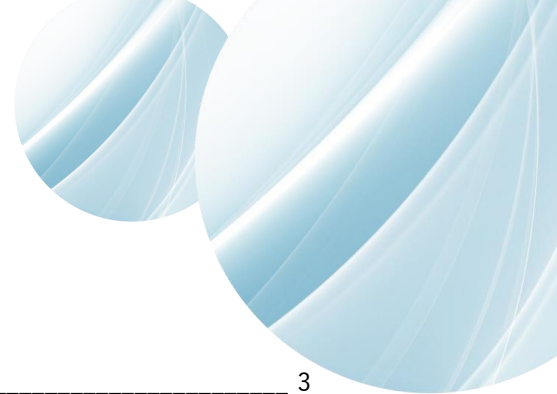


チーム開発演習
プロジェクト
説明資料



目次

| | |
|-----------------|----|
| 1 演習概要 | 3 |
| 1-1 演習の目的 | 3 |
| 1-2 演習における前提条件 | 3 |
| 1-3 本資料の構成 | 3 |
| 1-4 本資料の意図 | 3 |
| 2 プロジェクト概要 | 4 |
| 2-1 プロジェクトの目的 | 4 |
| 2-2 プロジェクトの位置づけ | 4 |
| 2-3 プロジェクトの期間 | 5 |
| 2-4 作業範囲 | 6 |
| 2-5 作業範囲外 | 7 |
| 2-6 成果物 | 8 |
| 2-7 会議体 | 9 |
| 2-8 資料提供 | 10 |
| 2-9 制約条件 | 11 |
| 3 演習の進め方 | 12 |
| 3-1 主な作業項目と手順 | 12 |
| 3-2 「1.体制整備」の手順 | 13 |

| | |
|--------------------------|----|
| 3-3 「2.仕様理解とコンセプト検討」の手順 | 14 |
| 3-4 「3.作業計画」の手順 | 16 |
| 3-5 「4.実装」の手順 | 16 |
| 3-6 「5.テスト」の手順 | 17 |
| 3-7 「6.成果報告」の手順 | 18 |
| 3-8 「オリジナル機能」の注意事項 | 19 |
| 4 環境 | 20 |
| 4-1 利用ソフトウェア | 20 |
| 4-2 成果報告会の環境 | 20 |
| 5 演習規約 | 21 |
| 5-1 演習を進める上での規約 | 21 |
| 5-2 作業分担における規約 | 21 |
| 5-3 ソースコードについての規約 | 21 |
| 5-4 コーディング規約・命名規則についての規約 | 21 |
| 5-5 画面レイアウトについての規約 | 21 |
| 5-6 利用する画像ファイルについての規約 | 22 |

1 演習概要

1 - 1 演習の目的

当研修内容を通して理解度、実装力がどれだけ向上したかを測ることを目的とする。

1 - 2 演習における前提条件

下記の前提条件を踏まえて、演習を行う。

- ・ 演習において、各個人はソフトウェア開発を請け負う一つのプロジェクトである
- ・ 期間内に成果物を納品する
- ・ 演習は納期や研修内容を考慮して、実用的なシステムの機能の一部を簡略化している

1 - 3 本資料の構成

本資料は、下記の構成となっている。

- ・ プロジェクト概要:各個人の立場とプロジェクトの責任範囲の説明
- ・ 演習の進め方:演習の進め方に関する説明

1 - 4 本資料の意図

本資料は、チーム開発演習の内容と演習の進め方を説明するための資料である。

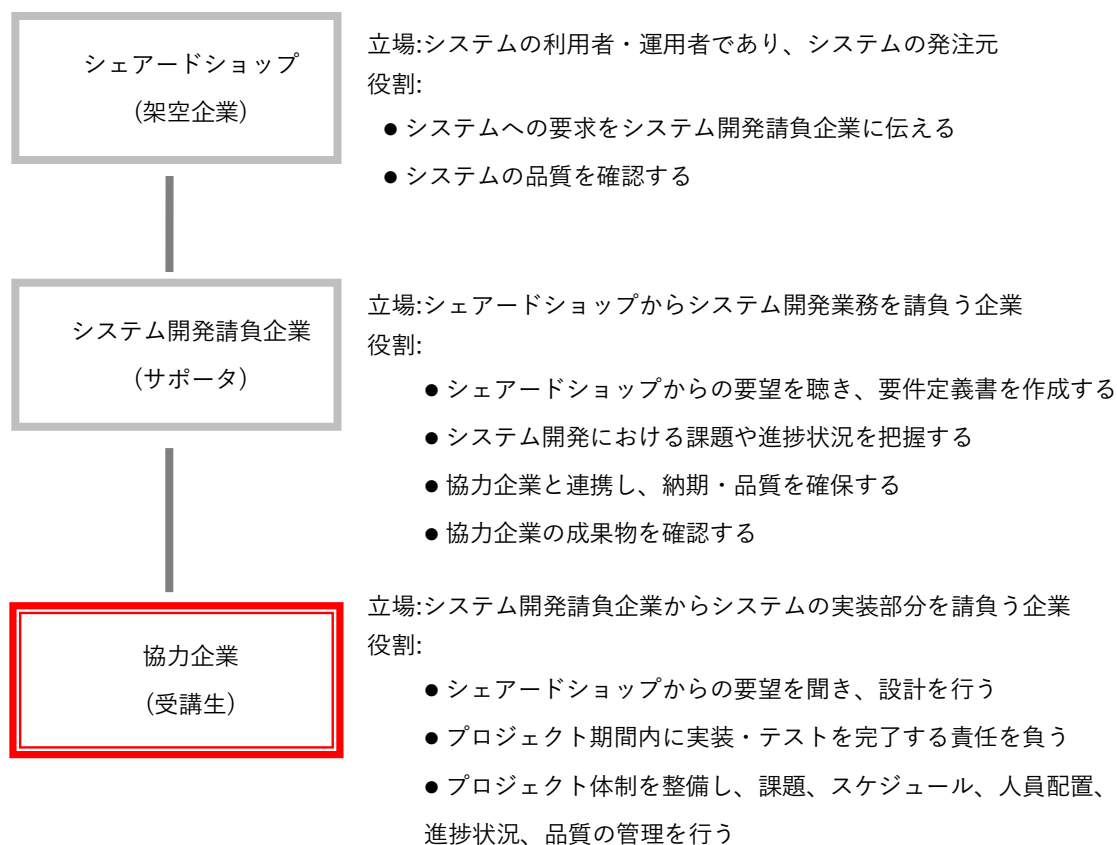
2 プロジェクト概要

2-1 プロジェクトの目的

シェアードショップの商品管理システムを通販システムへと改良することを目的とする。

2-2 プロジェクトの位置づけ

(1)組織的な位置づけ



以下、文中でシステム開発請負企業を「発注者(=サポータ)」、協力企業を「受注者」と表記する。

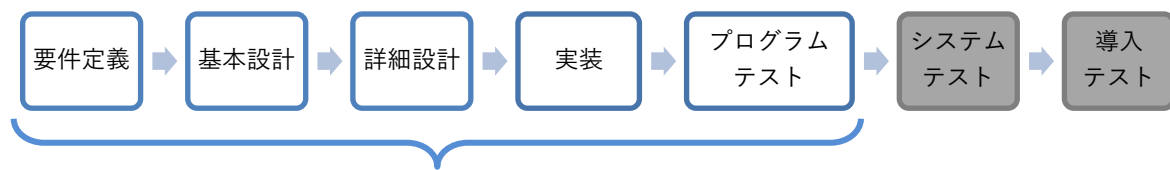
<現在の状況>

発注者から企業様に対して、要件定義書の内容を提示し、了承を得て、基本設計書、詳細設計書まで作成が完了している状況。

企業様から「独自性を持たせたい」という要望があり、新たに要件定義を行う必要がある。

<今後の予定>

既存の要件定義書、基本設計書、詳細設計書の内容を踏まえて、新たな機能について要件を定義し、発注者の了承を得たのち、基本設計、詳細設計を進める必要がある。



担当工程

2-3 プロジェクトの期間

プロジェクトの期間は研修コースにより異なるが、概ね以下のようなスケジュールで進める。



2 - 4 作業範囲

(1)必須作業:既存の仕様書内容の実装、テスト

下記の仕様書に従い、非会員・一般会員向け機能の実装を作業範囲とする。

- ・要件定義書(変更・追加分も含む)
- ・基本設計書(変更・追加分も含む)
- ・基本設計書_画面遷移(変更・追加分も含む)
- ・基本設計書_画面詳細(変更・追加分も含む)
- ・詳細設計書(変更・追加分も含む)

なお、下記の表に記載されている Controller クラスについては、既存のクラス(ソースファイル)に処理を追加することにより実装するものとする。

| 分類 | 機能名 | 既存ソースファイル名 (Controller パッケージ) |
|--------|----------|--|
| 1 汎用 | 1 ログイン | jp.co.sss.shop.controller.login.LoginController.java |
| 2 商品管理 | 1 商品一覧表示 | jp.co.sss.shop.controller.item.ItemShowCustomerController.java |
| | 2 カテゴリ検索 | |
| | 3 商品詳細表示 | |

また、下記のパッケージについては、非会員・一般会員機能の実装に合わせて、新規クラスの追加および既存クラスの改変(フィールドやメソッドの追加・変更)を行うこと。

| パッケージ名(アルファベット順) | 利用用途 |
|------------------------------------|-------------------|
| jp.co.sss.shop.annotation | 独自アノテーション |
| jp.co.sss.shop.bean | JavaBeans |
| jp.co.sss.shop.config | Spring Boot の設定 |
| jp.co.sss.shop.controller.basket | 買い物かご機能用コントローラ |
| jp.co.sss.shop.controller.category | カテゴリ管理 各機能用コントローラ |
| jp.co.sss.shop.controller.item | 商品管理 各機能用コントローラ |
| jp.co.sss.shop.controller.login | ログイン管理 各機能用コントローラ |
| jp.co.sss.shop.controller.order | 注文管理 各機能用コントローラ |
| jp.co.sss.shop.controller.user | 会員管理 各機能用コントローラ |
| jp.co.sss.shop.entity | エンティティ |
| jp.co.sss.shop.filter | フィルタ、セッションチェック |
| jp.co.sss.shop.form | 入力フォームの処理 |
| jp.co.sss.shop.repository | リポジトリ |
| jp.co.sss.shop.util | 共通処理、ビジネスロジック |
| jp.co.sss.shop.validator | 入力値のチェック処理 |

(2) 必須作業: ショップコンセプトの提案

顧客要望を考慮して、ショップを利用する年齢層や取り扱い製品、イメージカラーなどを決定し、要件定義として明確にすること。

(3) 追加作業: オリジナル機能の設計および実装、テスト

既存の仕様書に加えて、顧客要望を考慮して、追加機能や機能改善を検討し仕様書(変更・追加分)を作成し、実装、テストを完了すること。

オリジナル機能の仕様書は「要件定義書(変更・追加分)(ひな形)」「基本設計書(変更・追加分)(ひな形)」「基本設計書_画面遷移図(変更・追加分)(ひな形)」「基本設計書_画面詳細(変更・追加分)(ひな形)」「詳細設計書(変更・追加文)(ひな形)のファイルを使用してそれぞれ作成すること。

オリジナル機能の条件:

- ・データベースに新たなテーブルを追加すること
- ・一般会員限定の機能を追加すること
- ・管理者機能の改修は含めないこと

(4) 追加作業: AWS 環境へのデプロイ

最後の成果発表会では、作成したアプリケーションを AWS 環境へデプロイしてデモンストレーションを行うこと。

2-5 作業範囲外

下記の作業については、作業範囲外とする

- ・現行システムにおける不具合の修正
- ・システム構成、クラス構成、パッケージ構成、ファイル構造の変更
- ・システム設計、システムテスト、導入テスト
- ・運用・システム管理者機能用 Controller パッケージのクラス(クラス名に「Admin」を含む)の変更

2 - 6 成果物

(1) 成果物の種類

下記の成果物をプロジェクト期間内に作成し、発注者に納品する。

オリジナル機能を作成した場合は、変更・追加分として関連する仕様書も納品する。

| | 成果物 | 成果物の内容 |
|----|--------------------|--|
| 1 | 作業工程表 | 作業の担当者、作業の予定と実績状況が把握できるもの |
| 2 | 要件定義書(変更・追加分) | 顧客からの要望、受注の背景、システム構成、機能要件等を記載 |
| 3 | 基本設計書(変更・追加分) | ユーザインターフェースや各機能、データベース定義について記載 |
| 4 | 基本設計書_画面遷移(変更・追加分) | 画面遷移について記載 |
| 5 | 基本設計書_画面詳細(変更・追加分) | 各画面の詳細説明を記載 |
| 6 | 詳細設計書(変更・追加分) | パッケージ構成やテンプレート名、URL、プログラム構成について記載 |
| 7 | コーディング規約 | プロジェクトにおけるコーディング規約を記載 |
| 8 | テスト仕様書兼成績書 | テスト項目、テスト実施者、実施日、テスト結果を記載 |
| 9 | ソースコード | システム全体の Java ソースファイルおよびテンプレート(HTML ファイル)、SQL、関連するプロパティファイル 等 |
| 10 | 課題管理表 | プロジェクトにおける課題の一覧、解決策も含む |
| 11 | 成果報告資料 | 成果報告時に使用するプレゼンテーション資料 |

(2) 成果物の形式

- ・ 成果物は、全てデジタルデータとして作成する
- ・ ソースコード以外の成果物については、任意のフォーマット、ファイル形式で作成する

(3) 納品完了の判断基準

- ・全ての成果物がそろっていること。
- ・全ての成果物に作成者名、作成日、更新者名、更新日が明記されていること
ただし、画像ファイル、現行システムのソースコードについては対象外とする
- ・テストが完了していること
- ・残存バグ^{※1}や不具合^{※2}が課題管理表に明記されていること

※1 残存バグ:ここでは、プログラミングを原因とする動作異常のこととする。

テストにおいて、設計書と異なる動作をすることは認識できているが修正できていないもの。

※2 不具合:ここでは、設計上の問題、環境上の問題による動作異常のこととする。

テストにおいて、PCの違い、操作のタイミングにより設計書とは異なる動作をすることが認識できており、プログラムコードが原因ではないと判断しているもの。

2-7 会議体

下記の内容に従って、受注者が主導し、会議を開催する。

| 名称 | 参加者 | 目的 | 開催頻度 |
|---------------|------------|--|--------------------|
| 進捗報告会 | 発注者 受注者 | 作業工程表を用いて、発注者にプロジェクトの状況を伝え、工程の見直しの必要性を判断する。 | 毎日 |
| 課題管理表 レビュー | 発注者 受注者 | 課題の内容、解決状況を発注者に伝え、残課題についての方向性を決定する。 | 課題管理表変更時 |
| 作業工程表 レビュー | 発注者 受注者 | 発注者に作業工程の内容を伝え、抜けや漏れがないかを確認する。 | 作業工程表作成 および、変更時 |
| チーム内会議 | 受注者 | チーム内の各自の作業状況や課題、進捗状況を把握し、作業割り当て、スケジュールの見直しを行う。 | 任意 |

2 - 8 資料提供

システム開発に必要な下記資料及びデータは発注者より提供する。

要件定義書

- ・お客様の要求を基に発注者がまとめた資料
- ・発注者側で確定済みの要件が記載されている
- ・受注者側で検討したオリジナル機能については、要件定義書（変更・追加分）を作成する

基本設計書

- ・発注者がシェアードショップ様に対してシステムを説明するために作成した資料
- ・ユーザインターフェースや各機能、データベース定義について記載
- ・発注者側で確定済みの仕様が記載されている
- ・受注者側で検討したオリジナル機能については、基本設計書（変更・追加分）を作成する

詳細設計書

- ・発注者が基本設計書に基づいて作成したソフトウェアの設計書
- ・パッケージ構成やテンプレート名、URL、プログラム構造について記載
- ・発注者側で確定済みの仕様が記載されている
- ・受注者側で検討したオリジナル機能については、詳細設計書（変更・追加分）を作成する

現行システムのソースコード

- ・シェアードショップ様商品管理システムのソースコード

HTML モック

- ・画面レイアウトのサンプル(発注者から参考資料として)
- ・Thymeleaf のテンプレートファイルを作成するための素材(HTML,CSS,画像)
- ・受注者は設計した仕様に沿って HTML モックを適宜改編することが許可されている

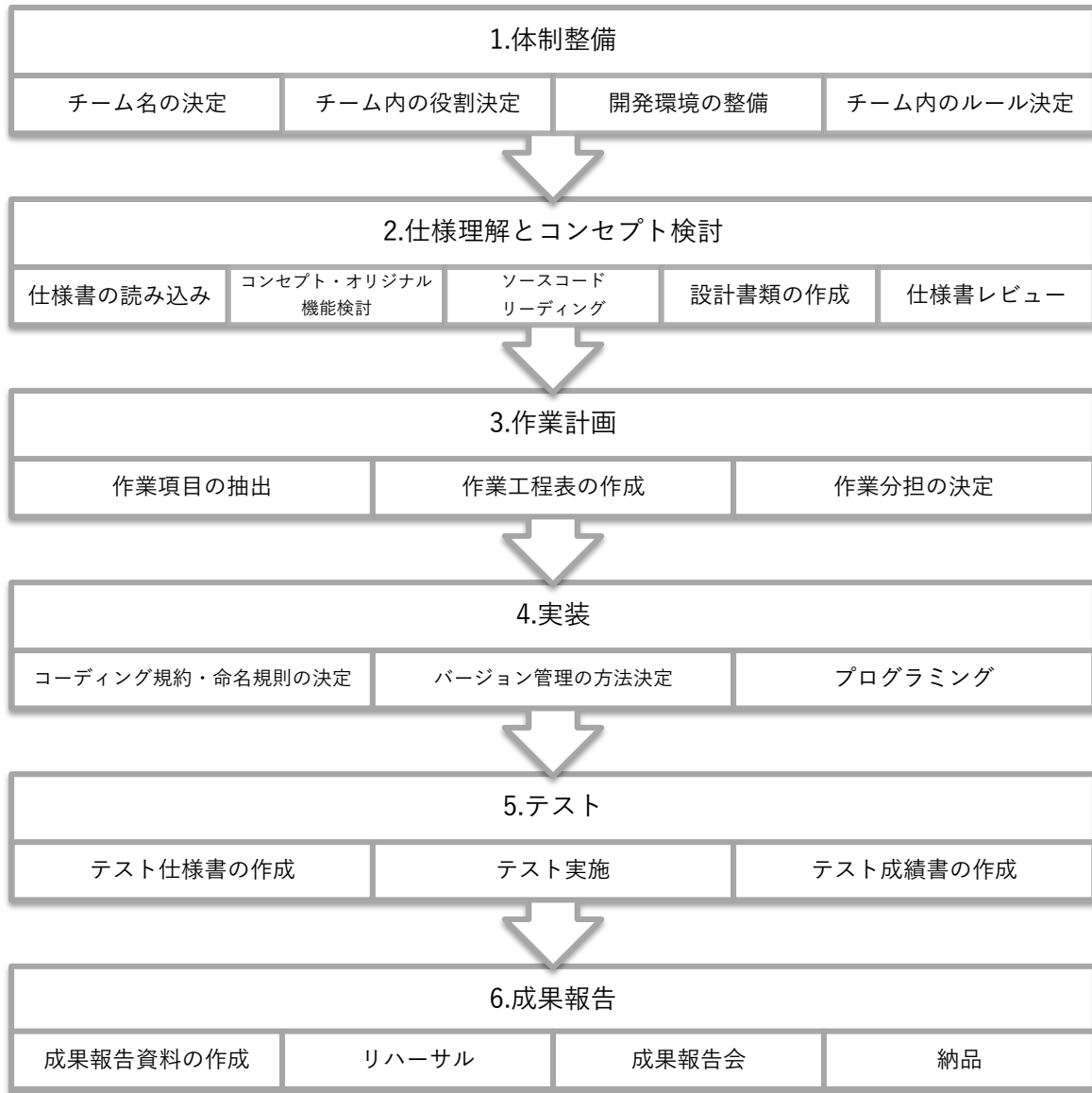
2 - 9 制約条件

- ・ 本プロジェクトに関連する資料及び成果物の著作権は、発注者に帰属する
- ・ 設計資料及び開発対象外のソースファイルを無断で改変すること(コメント追記も含む)を禁止する
- ・ 本番環境での動作検証ができないため、ローカル環境での動作確認および、最終的には AWS 環境での動作確認を行う
- ・ ローカル環境での動作確認には、下記の URL を使用すること
`http://localhost:55000/shared_shop/`

3 演習の進め方

3-1 主な作業項目と手順

下記の順で演習を進める。



3-2 「1.体制整備」の手順

(1) チーム名の決定

チームの名称を決定する。

※チーム名は他のチームと重複しないこと

※チーム名は成果報告会で使用する発表資料にも載せる名前である

そのため、他者を不快にする名前、公序良俗に反する名前は付けないこと

(2) チーム内の役割決定

チーム内で下記の役割に対して、担当者を割り当てる。

※チームの人数が4名以下の場合は、1名が2役以上を兼任すること。

| 役割名 | 役割 | 関連する成果物 |
|-----------|--|---|
| リーダー | <ul style="list-style-type: none"> ・ チーム全体の統括管理・推進を行う。 ・ 各担当者から情報を収集し、状況を発注者へ報告する。 | 全ての成果物 |
| 課題管理者 | <ul style="list-style-type: none"> ・ 納期、品質、作業量など期間内の作業完了に影響を与える可能性のあることを課題として管理し、解決策の検討を推進する。 <p>例 1: ○○処理の実現方法を調べる必要がある。</p> <p>例 2: △△機能のプログラミングに想定していたより作業に時間がかかる。</p> | 課題管理表 |
| スケジュール管理者 | <ul style="list-style-type: none"> ・ 作業項目の抽出、作業の進捗状況の管理を行う。 ・ 作業工程表の作成と見直しを行う。 | 作業工程表 |
| 人員管理者 | <ul style="list-style-type: none"> ・ どの作業を誰が行うかを割り当てる。 ・ スケジュール管理者、課題管理者と連携し、作業や課題解決に抜けや漏れが起きないようにする。 | 作業工程表 |
| 品質管理者 | <p>システム開発プロジェクトの成果物に対する下記の品質基準を定め、維持できるように推進する。</p> <ul style="list-style-type: none"> ・ コーディング規約 ・ 命名規則 ・ テスト仕様書、成績書の記載方法 | <p>コーディング規約</p> <p>命名規則</p> <p>テスト仕様書</p> <p>テスト成績書</p> |

(3)開発環境の整備

各自の PC にて、下記の作業を行い、環境を整備する。

- ・仕様書、HTML モック、現行機能のソースコードを準備する。
- ・ eclipse へのソースコード、テンプレートファイルをインポートする。
- ・バージョン管理ソフトの設定と使い方を理解する。
- ・ eclipse でプロジェクトを作成しておく。
- ・プロジェクト名は、「shared_shop_チーム名(英語表記)」とする。

(4)チーム内ルールの決定

チーム内のコミュニケーション・情報共有ルールについて、下記の点を考慮し決定する。

- ・いつ、どのような情報、どのような目的で、誰と誰が共有するのか(5W1H)
- ・「2-7 会議体」を考慮し、発注者への報告を誰がどのように行うのか
- ・ソースコードなどの成果物をどのように共有するのか
- ・課題管理表に記載する条件、課題完了判断の条件はどうするのか

※以降、テスト完了までのフェーズにおいては、不明点・疑問点・不具合について、課題管理表にその内容と解決方法・対応内容を記載し、管理すること。

3-3 「2.仕様理解とコンセプト検討」の手順

(1)仕様書の読み込み

下記の順で資料を読み、システムの全体像を把握する。

- 1.「顧客要望まとめ」
- 2.「要件定義書」
- 3.「基本設計書」 画面遷移図、画面詳細を含む
- 4.「詳細設計書」

(2)コンセプト・オリジナル機能検討

- ・ショップのコンセプトと追加するオリジナル機能をチーム内で検討し、決定する

※ 3-8 「オリジナル機能」の注意事項 を参照すること

(3)ソースコードリーディング

既存のソースコードを読み、下記の点を理解・決定しチームで共有する。

- ・ソースコードの構成を理解する
- ・新規作成・変更するソースファイルを決定する
- ・新規作成・変更するテンプレートファイルを決定する
- ・その他関連するプロパティファイル等を決定する

(4)設計書類の作成

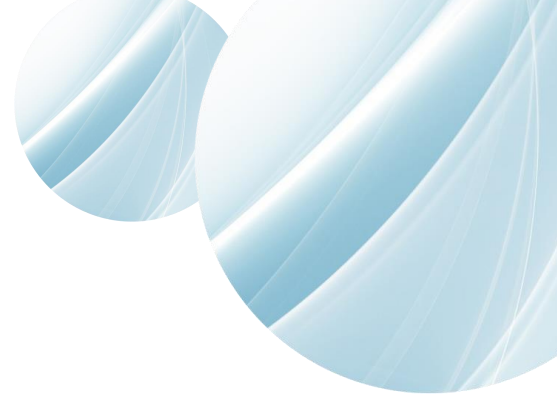
・上記(2)(3)を踏まえて、
「要件定義書」→「基本設計書」→「詳細設計書」
の順にチーム内で相談し、変更・追加分の設計書類を作成する。

(5)仕様書レビュー

上記(4)にて作成した設計書類のレビューを行い、不具合等の検出・修正を行うとともに、チーム内での認識合わせを行う。

レビューは下記の手順で実施する。

- 手順1. 各仕様書の追記内容について発注者にレビューを依頼する
- 手順2. 発注者から指摘を受けた場合、指摘事項に沿って仕様書を修正する
- 手順3. 修正完了後、修正箇所について発注者に再度レビューを依頼する



3-4 「3.作業計画」の手順

(1)作業項目の抽出

下記の手順で作業項目を抽出する。

手順 1.各成果物を作成するための手順を細かく検討し作業項目一覧をつくる

手順 2.作業項目ごとの完了条件を決める

手順 3.作業項目に抜けがないか確認する

(2)作業工程表の作成

抽出した作業項目の作業順序を決定し、工程表としてまとめる。

(3)作業分担の決定

作業項目 1 つに対して担当者を 1 人以上決定し、下記の点を確認する。

- ・作業順序に不整合がない
- ・作業分担に偏りがない
- ・各作業項目の着手予定日、完了予定日が明確になっている
- ・作業完了判断は、誰が、いつ、どのように行うかが明確である

3-5 「4.実装」の手順

(1)コーディング規約・命名規則の決定

- ・コメントの記述方法、記述内容、インデント位置、改行位置などのコーディング規約を決定し、文書としてまとめる
- ・クラス名、メソッド名、フィールド名、ローカル変数名、定数名などについて、命名規則を決定し、文書としてまとめる
- ・コーディング規約の模範となる書き方は、講義資料『Java_講義資料(下巻)』を参照する

(2)バージョン管理の方法決定

バージョン管理ソフトをいつ、どのように利用するかをチームで決定する

(3)プログラミング

- ・ 決定した作業工程表、命名規則、コーディング規約に従い、プログラミングを進める
- ・ 各画面はHTML モック、および「基本設計書_画面詳細.pdf」を基に作成する
- ・ 品質管理者は、命名規則、コーディング規約に則っているかを確認する
- ・ スケジュール管理者は、進捗状況を把握し、予定より進んでいるもしくは遅れている場合
人員管理者と相談し、作業分担を見直す

3-6 「5.テスト」の手順

(1)テスト仕様書の作成

下記の手順でテスト仕様書を作成する。

手順 1.基本設計書、詳細設計書を元に確認する項目を抽出する

手順 2.確認するための手順、確認時に入力する値を明確にする

手順 3.テスト結果としてどのような結果が正常かを明確にする

手順 4.テスト仕様書としてまとめ、項目に抜けがないかを確認する

(2)テスト実施

- ・ テストに必要なデータ(画像データ、データベース内のデータ含む)を準備する
- ・ テスト仕様書に基づいて、テストを実施し、記録に残す
- ・ 画面のイメージが必要な場合は、スクリーンショットを記録として利用する
- ・ 不具合※が見つかった場合もテストを中断せず、最後までテストを行った後、不具合の原因調査を行う
- ・ 不具合が発生した場合は、設計書で仕様を確認の上、必要に応じて設計書およびソースコードを修正する
- ・ 不具合箇所については、プログラム修正後、再度テストを行う。

※不具合:ここでは、想定していたテスト結果と異なる状態のことをいう

(3)テスト成績書の作成

- ・ テストの実施記録を成績書としてまとめ、テスト漏れがないことを確認する
- ・ 不具合の原因が明確にならないもの、同じ手順で行っても現象が再現できないものについては課題管理表に記録する

3 - 7 「6.成果報告」の手順

(1)成果報告資料の作成

成果報告資料のサンプルデータを元に、チームの成果をまとめる。

(2)リハーサル

成果報告のプレゼンテーション、発言する順番などを決定し、リハーサルを行う。

(3)成果報告会

チームごとに成果報告を行う。

(4)納品

納品作業として、下記の作業を実施する。

- ・全ての成果物をリポジトリ上にコミットし提出する
- ・コミット後にサポータに伝え、成果物の確認を依頼する

3-8 「オリジナル機能」の注意事項

(1)仕様書への記述について

- ・要件定義書、基本設計書、詳細設計書の変更・追加分を作成すること
- ・テスト仕様書にオリジナル機能分の記述を行うこと
- ・作成したドキュメント類は、必ず発注者にレビュー依頼をし、完了判断を仰ぐこと

(2)実装について

オリジナル機能は、必須作業であるため、1 つ以上の機能を追加実装すること

(3)制約

- ・顧客要望※とかけ離れた機能の実装は再考すること ※「顧客要望まとめ」を参照
- ・オリジナル機能のための、DB テーブル変更、追加を許可する

4 環境

4 - 1 利用ソフトウェア

- ・下記のソフトウェアを利用する
- ・下記のソフトウェア以外のものを利用したい場合は、発注者へ相談する

| ソフトウェアの名称 | ソフトウェアの種類 |
|---|------------------|
| Java(Oracle Java SE Development Kit 16) | Java 開発キット |
| Spring Boot 2.7.x | Spring フレームワーク |
| Oracle Database 21c Express Edition | DBMS |
| Eclipse 202209 | 統合開発環境 |
| Tomcat 9 | Java アプリケーションサーバ |
| Windows 10 | OS |
| Libre Office などのオフィスソフト | ドキュメント作成アプリケーション |
| SAKURA Editor | テキストエディタ |
| GitHub | バージョン管理 |
| TortoiseGit | Git クライアント管理ツール |

4 - 2 成果報告会の環境

- ・成果報告会で行うプレゼンテーション、デモンストレーションの動作確認を事前に行っておくこと
- ・成果報告会では、AWS 環境へデプロイしデモンストレーションするものとする
- ・その際の URL は、下記のものを使用すること

[http://\[Web アプリケーション用 EC2 の IP アドレス\]:55000/shared_shop/](http://[Webアプリケーション用 EC2 の IP アドレス]:55000/shared_shop/)

5 演習規約

5-1 演習を進める上での規約

- ・ 不明な点がある場合、基本的にはチーム内で相談し解決すること
- ・ チーム内での解決が難しい場合は、サポータに相談すること
- ・ 他チームに相談する必要がある場合は、必ず相談先チームのリーダーに許可をとること

5-2 作業分担における規約

- ・ 各自が必ず作業項目における 1 項目以上の Java プログラミング作業を受け持つこと
- ・ 習得状況により、作業分担量を調整し、開発期間内に作業を完了させること

5-3 ソースコードについての規約

- ・ 開発対象範囲外のソースコードは、編集しないこと
- ・ 既存のソースコードを編集する場合は、変更箇所がわかるようにすること
- ・ 原則としてパッケージの構成は既存の構成に従うこと

5-4 コーディング規約・命名規則についての規約

- ・ 全てのクラス、全てのメンバ（メソッド、フィールド、定数など）に、Javadoc 形式のコメントを記述すること
- ・ ただし、インターフェイス内のメソッドに Javadoc が記載されている場合、実装クラス内でオーバーライドしたメソッドの Javadoc は省略可能とする
- ・ クラス名、メソッド名、フィールド名については、既存ソースコードと同等の命名規則に従うこと

5-5 画面レイアウトについての規約

- ・ 各画面のレイアウトは HTML モックを参考として、各自で検討してアレンジすること
- ・ HTML モックのレイアウトをそのまま流用することも許可する

5 - 6 利用する画像ファイルについての規約

当該システム上で使用する画像データは、以下のみ許可する。

- ・ 商用での無償利用、改変許可がされている画像データ
- ・ パブリックドメインの画像データ
- ・ ペイントツールで自作した画像