

システムプログラミングⅡ・レポート課題指示書 自作サービスの設計・実装

2025 年度（R7）

2025-07-30 案公開

2025-10-29 概要設計プレゼン明記、誤字脱字やレイアウト修正

システムプログラミングⅡの後期はサービスを設計・実装する課題を与える。この文章はその指示書である。矛盾や疑問を見つけた場合は早めに申し出て欲しい。

1 目 的

シラバスにあげた「ネットワークを利用するプログラムを作成できる」「データベースを利用するプログラムを作成できる」という目標を達成するため、自作プログラムを含めたサービスを設計・実装する。

2 概 略 手 順

- 1) 各自で作成するサービスの概略設計を行い、プレゼンテーション発表する。3.1 節参照。
- 2) 1) の詳細設計を行う。3.2 節参照。
- 3) 2) の実装を行う。3.2 節参照。設計との齟齬があれば、設計修正も行う。
- 4) 3) の動作確認を行う。設計上の問題があれば 2) へ、実装上の問題があれば 3) へ戻って修正する。3.3 節参照。
- 5) 自作したサービスやサーバの紹介・デモンストレーションを実施する。3.4 節参照。
- 6) 自作したサービスの説明、プログラム解説を記述したレポートをサービス構築・運用に必要なファイル群と共に提出する。3.5 節参照。

3 条 件

ここでは、各種条件を示す。

3.1 サービス条件

図 1 にサービス概念図を示す。続けてサービス条件を述べる。

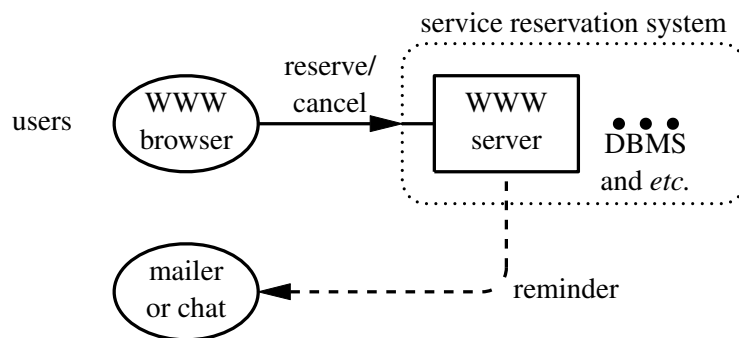


図 1: 予約サービス構成概念図

- 対象は WWW を使ったなんらかのサービスの予約システムとする。例えば、航空券の予約、会議室の予約、図書借用の予約など。以下、このなんらかのサービスを「本来サービス」と呼び、予約システムの方を「予約サービス」と呼ぶ。
- 予約時間が近づくとメールあるいはチャットなどで通知を届ける（以下「リマインダー」と呼ぶ）こと。リマインダーを発行する時間はサービス内容に応じて調整する。例えば、「二日目前にリマインダーが届く」「15 分前にリマインダーが届く」など。

- 一人以上の管理者と複数人の一般利用者が利用できること。
- 各利用者が予約一覧と個別予約の内容を閲覧可能なこと。
- 管理者は全ての予約のキャンセル、予約サービスの停止を実施できること。
- 管理者は本来サービスが実施できない際には、該当する全ての予約に対して、本来サービスが実施できない旨の通知（以下、「本来サービス実施不能通知」）をすること。例えば、航空券予約では悪天候で運行がキャンセルされることがある。会議室や図書館でも停電その他で利用できないことがある。
- 管理者およびそれ以外の利用者（一般利用者）はパスワードで認証する。
- この課題では原則として料金に関する項目は問わない。料金はオプションな加点項目とする（4 節）。

* * *
- サービスやシステムの概略を教員・他学生の前でプレゼンテーションする。プレゼンテーションの詳細（日時、発表順など）は別途アナウンスする。

3.2 設計・実装条件

- 予約や予約確認、予約キャンセルは HTTP を使うように設計すること。
- 中心部分は Rust を使って実装すること。ただし、部分的に得意とするシステムや言語をサブシステムとして使うことは妨げない。例えば、「データベース操作は SQL を使う」「サーバ起動はシェルを使う」など。
- 内部になんらかのデータベースを持つこと。DBMS の種類や実装を問わない。例えば RDB や KVS(Key-Value Store) などが利用できるだろう。
- データは永続的に保持すること。サーバを停止したら予約を失うようなシステムは認めない。
- 本来サービスの内容に応じて予約内容の詳細を決めること。
- 当初のリマインダー時間より短い時間の急な予約登録でもリマインダーは発行すること。例えば、15 分前にリマインダーを発行するサービスで、15:00 の予約を 14:50 分（10 分前）に登録されてもリマインダーを発行する。
- 利用者は最小限 10 人以上に対応するように設計すること。
- 通知方法のチャットは Microsoft Teams か Google Meet とする。
- 似たようなプログラムをそのまま使うようなシステムは認めない。
- 仮想マシンやコンテナを使っている場合、それらの起動・停止は自動化せずとも良い。

3.3 動作確認条件

- 全ての機能を表にして、それぞれを確認したことを示すこと。
 - 確認の根拠として作業メモや画面ハードコピーをとること。
- 特にリマインダーの動作は厳密に確認すること。
 - 実時間で動作確認すると大きな時間を消費するので、時刻を早めて確認してよい。ただし、時刻早めると動作に悪影響を与えることがあるので、各自のプログラムで早めることで悪影響を与えないことを確認した方がよい。

3.4 プレゼンテーション条件

- 一般利用者と管理者に関する一連の動作を見せること。
- 最低限一般利用者 2 人以上と管理者 1 人以上としての動作を見せること。
 - リマインダーと本来サービス実施不能の通知は必ず含めること。
- ★ ★ ★
- 開催時期は後期末とする。2-3 日かかると思われる。具体的な日付は人数に応じて調整後に周知する。
- 発表時間は一人 10-15 分程度とする。日程調整により変化する可能性がある。

3.5 レポート条件

- 提出物は全て学生番号で始まるファイル名をつけること。
- 表紙に題名、学生番号、氏名をつけること。
- レポートは PDF 形式とする。
- 対象とする本来サービスを処理の流れの図を添えて説明すること。
- 予約サービスのシステム構成（モジュールなど）の図をつけること。必要に応じて処理の流れや状態遷移などの図もつけること。
- データベースの構成（テーブルなど）を図に示すこと。
- 各種機能の紹介を含めること。
- 各種機能の動作確認の手順を含めること。
- 紹介や動作確認では項目ごとに画面ハードコピーを添えること（WWW ブラウザ上での予約、予約キャンセルなど。メーラやチャットプログラム上での通知内容など）。
- 反省や振り返りについても記述すること。
- 「まとめ」に相当する章や節を設けて記述すること。
- 参考資料（文献、WWW ページなど）を記述すること。
- 図や表を引用した場合は、その周りに出典を明記すること。
- ★ ★ ★
- 作成したプログラムを提出すること。
- プログラムそのものだけでなく、コンパイルや実行に必要なファイル（Cargo.toml、データベース関連ファイル、仮想マシンやコンテナ類設定ファイルなど）も含めること。
- これらプログラム関連ファイル群は上のレポートをまとめてもよい。
- ★ ★ ★
- Classroom の所定の場所に〆切までに提出すること。
- 〆切日はプレゼンテーション時期が近づいたらアナウンスする。

3.6 補 足

- 2 節の通り、各自、すなわち一人で設計・実装すること。
- データベース周りは先の通り RDB や KVS を勧めるが、テキストファイルや dbm やその派生などのライブラリの利用も認める。
- 開発・検証・デモの際に時刻加速のために OS の時間を変えることもあるだろう。OS 時刻変更を容易にするために仮想マシンを使うことを勧めるが、必須ではない。

4 加 点 項 目

以下のような機能を持つ場合、更新を実施した場合に加点する。ただし、見落とし防止のためにレポートではその旨を明記すること。これらは3.1 節の教員審査の対象としないので、随時取り組んで良い。

- 全ての利用者の全ての予約を一覧できる機能。
- 予約時間を過ぎると利用感謝のメッセージを送る機能。
- キャンセル待ちの機能。キャンセル待ちが成功・失敗した際には該当利用者へその旨通知する。
- 料金を含めたシステムに更新する。本来サービスが実施不能の際には払い戻しに対応すること。
 - － 料金の支払い、払い戻しはメールやチャットあるいはその両方での通知だけ実施されたとみなす。
- その他、利用者や管理者の利用を助ける機能

5 注 意 点

- この指示書に従わない場合、課題の提出を拒否する、あるいはやり直しを命ずることがある。
- 教員の審査を受けずに詳細設計や実装を行った場合は、やり直しを命ずることがある。
- べ切に間に合わないプレゼンテーションやレポートは成績をつけない。
- 要領を得ない文章は成績をつけない。
- 不正が発覚した場合は成績を取り消すことがある。
 - － 人工知能の作成物をそのまま提出してはならない
 - － 同じあるいは酷似したプログラムが見つかった場合、該当者全員の成績を取り消すことがある。