

Next 班
最終レポート

藤原育実 環境情報 3 年

野上大輔 環境情報 3 年

安藤亮一 環境情報 3 年

PM:佐々木雄生

目次

1. 概要	3
2. 業務分析	5
3. 要件定義	8
4. 設計・実装.....	15
5. 個人レポート	39

1. 概要

我々Next 班は、NextWare 社の勤務状況管理を改善するシステムを作成した。NextWare 社では、勤務状況管理において、月次報告と週次報告で二重入力が発生していたり、社員全員分のファイルを開いて確認しなくてはならなかったりするという問題が発生していた（詳細は業務分析の項を参照）。我々は、二重入力の問題を、月次報告で作成したデータを週次報告作成用のクライアントアプリケーションで読み込めるようにして解決した。社員全員分のファイルを開いて確認しなくてはならないという問題は、社員のデータを Web アプリケーションで一括して表示できるようにして解決した。

1.1. 目的

NextWare 社で発生している勤務状況管理で発生している

- ・ 月次報告と週次報告で二重入力が発生してしまう問題
- ・ 社員全員分のファイルを開いて確認しなくてはならない問題を解決すること。

1.2. 対象とするユーザ

- ・ クライアントアプリケーション … NextWare 社の従業員
- ・ Web アプリケーション … NextWare 社の管理職員
- ・ メール送受信アプリケーション … NextWare 社の管理職員

1.3. 開発環境・使用技術

- ・ 開発人数：3 人+PM 1 人
- ・ 開発言語：
 - クライアントアプリ…VBA
 - Web アプリ…Java
 - メール送受信アプリ…Java
- ・ VBA 開発ツール：Microsoft Excel
- ・ Java 開発ツール：Eclipse
- ・ データベース：PostgreSQL
- ・ Web アプリケーションコンテナ：Tomcat
- ・ メール送受信 API：JavaMail

1.4. 開発期間・開発目標

- ・ 2005 年 10 月開発開始
- ・ 最終発表日である 2006 年 2 月 9 日までに，要件を満たしたシステムを完成させる

2. 業務分析

2.1. 前提状況

予てより A 社に於いては、月次ベースの勤務状況管理が行われており、それに際して Microsoft Excel および紙の帳票が利用されていた。

このほど新たに、残業や休日出勤などの超過勤務について、各従業員が事前に予定を立てて上司に申請して承認を得る制度を開始した。ところが新制度の承認フローに必要な項目等が従来の管理帳票に含まれていなかったため、超過勤務報告用には追加の帳票を用いて週次ベースで管理を行う運びとなった。しかし従来の報告用の帳票と、追加の帳票内で重複する項目が多く、二重入力の手間が存在している。

また実際の週次報告段階については、従業員は入力済みの Microsoft Excel のファイルを上司に送信することで申請を行っている。送信にはメールを利用しているが、この際の添付ファイルの容量が自ずと肥大化していくために扱いづらいという問題がある。

また、上司は管理している部下の帳票を閲覧する必要があるが、現状では全員のファイルを一括して閲覧することが困難であり、全員分の報告を把握するために手間が掛かっている。また一括して従業員の勤務状況を閲覧できないことで状況の俯瞰的な把握が難しく、閲覧結果を分析するに最適とは言えない状況である。

本プロジェクトの目的はこれらの事態を改善することにある。

2.2. 前提制約

既知の問題として A 社には社外用のサーバが存在しないことがある。即ち、システムに Web を利用する場合はそのアクセスは社内限定されることとなる。

しかし、A 社はその勤務形態により客先常駐社員が存在するため、そうした社員が直接システムにアクセスすることは難しい。客先によっては Web を利用できない環境にある可能性もあり、さらに自宅でもインターネットを利用できない環境にいる社員が存在するため、全社員が必ずしも Web を利用できる環境にあるとは限らない。こうした社員はごく一部であることが判明しているが、確実に存在している。

同様に、客先常駐社員が利用できるシステム環境が完全に客先によることもあり、例えば特殊な環境が必要なシステムの場合はすべての客先で利用できるかは不明である。

新システムの条件として、その問題を解消、あるいは代替案を考慮することが必須である。

2.3. ステークホルダー

本プロジェクトに関与するA社のステークホルダーは、次の4者である。

- ・ 従業員
- ・ エリア管理者
- ・ 管理責任者
- ・ 事務担当者

従業員は、社内勤務あるいは客先常駐という勤務形態を取り、自らの勤務状況（次週の予定および先週の実績）をエリア管理者に報告し、承認を受ける。

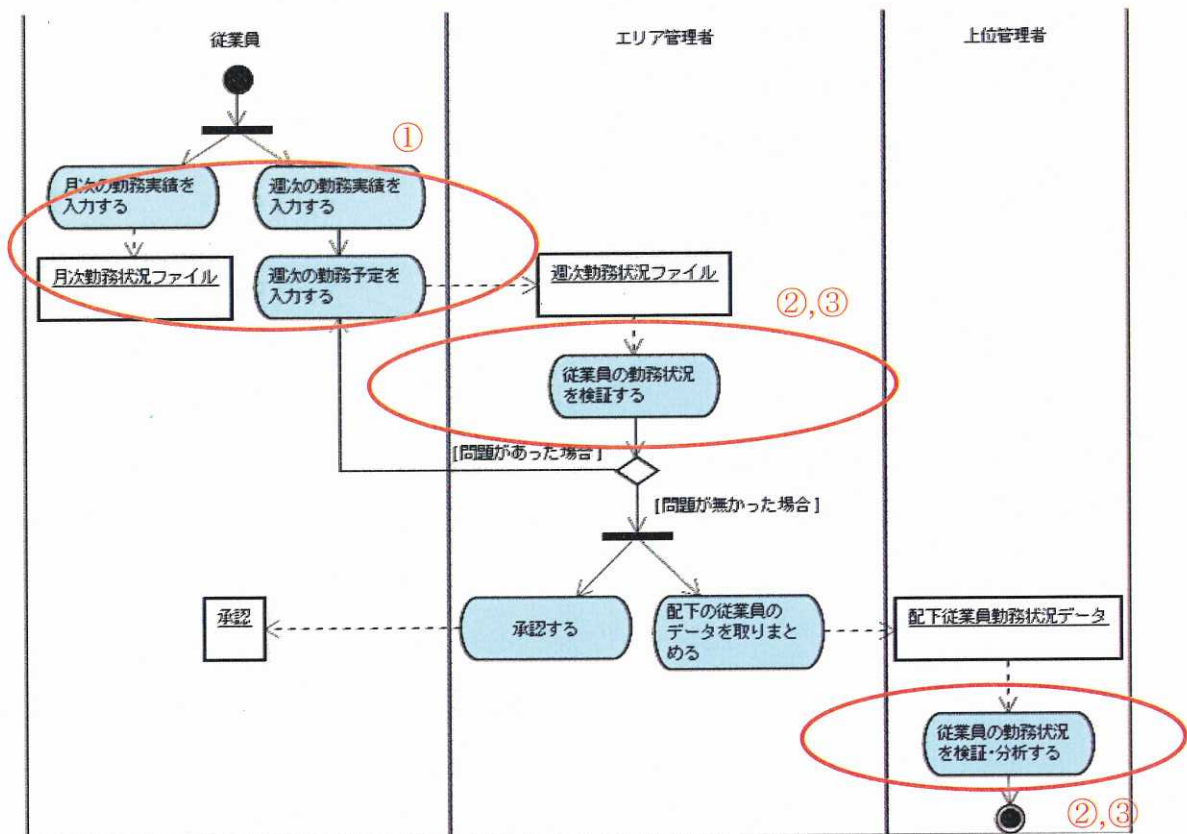
エリア管理者は、配下の従業員あるいは下位のエリア管理者の報告を受けてその状況を判断し、従業員に対して改善を指示あるいは客先に対して交渉を行う。例えば判断材料として、過度の超過勤務などの適切でない勤務状況や、あるいは勤務予定と勤務実績との大幅な齟齬などを分析する。またエリア管理者は、配下の従業員の勤務状況をとりまとめて上位のエリア管理者あるいは管理責任者に報告を行う。

管理責任者は、エリア管理者からの報告を元に、その責任範囲である部門の勤務状況を上位から分析し、コスト最適化のための経営的な判断を行う。例えば、プロジェクトの多忙さなどから人員の分散を再検討する。

事務担当者は、勤務実績をもとに給与計算等の事務的处理を行う。また、諸事情によって指定の形式で報告できない従業員の勤務状況報告書を代理作成する、といった諸雑務を処理する。

本プロジェクトの成果は、上記各ステークホルダーのいずれに対しても過度の負担を強いることなく、A社における関連業務全体の作業量を低減するものでなければならない。

2.4. 現行運用フロー



[現行運用フロー問題点]

現行の運用フローの問題点は以下の部分である。

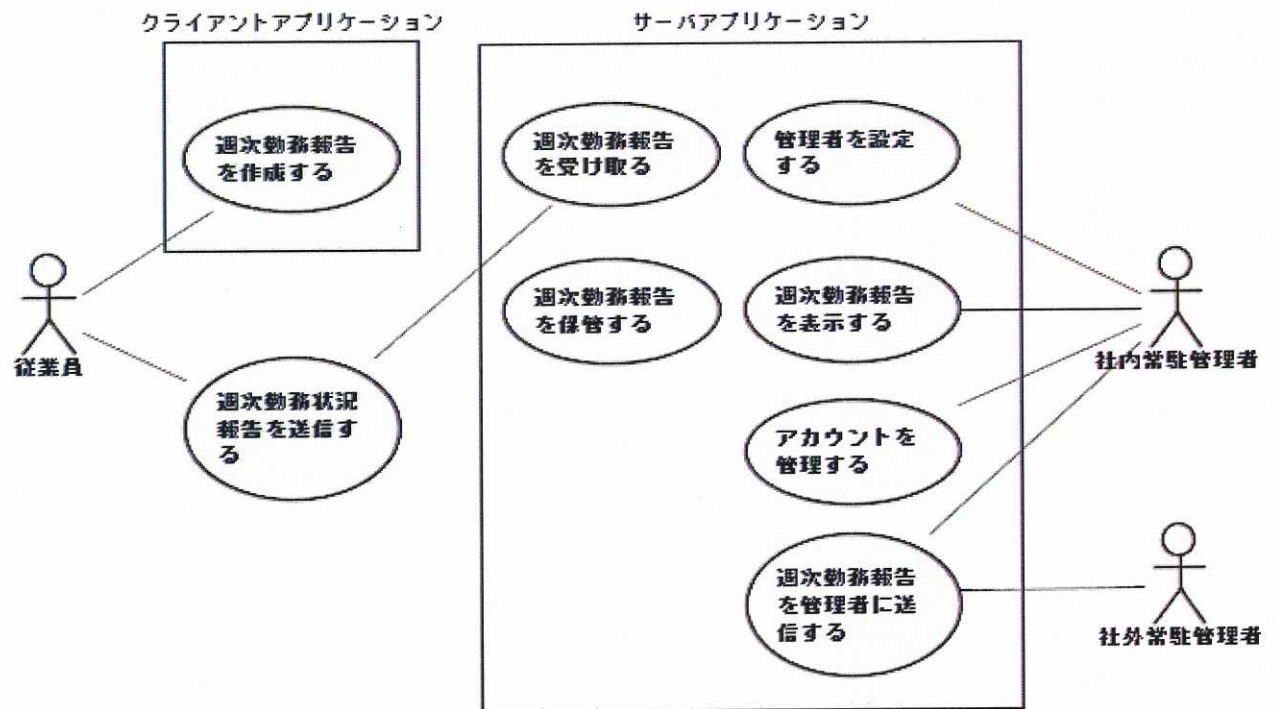
- ① 従業員は、自分の勤務状況報告を入力する際、月次報告と週次報告の両方に勤務実績を入力しなければならないという、2重入力が発生している。
- ② 従業員の勤務状況報告を受け取ったエリア管理者や上位管理者は、従業員からの報告ファイルを一つ一つ開いて検証していかなければならず、手間がかかる。
- ③ エリア管理者や上位管理者は、従業員の勤務状況を一覧で閲覧できないため、日付ごとの予実など細かい部分を検証できず、実態として予実の合計部分しかチェックできていない。
- ④ データが分散して存在しているため、上位管理者による分析が事実上不可能である。

3.要件定義

3.1. システム化要件

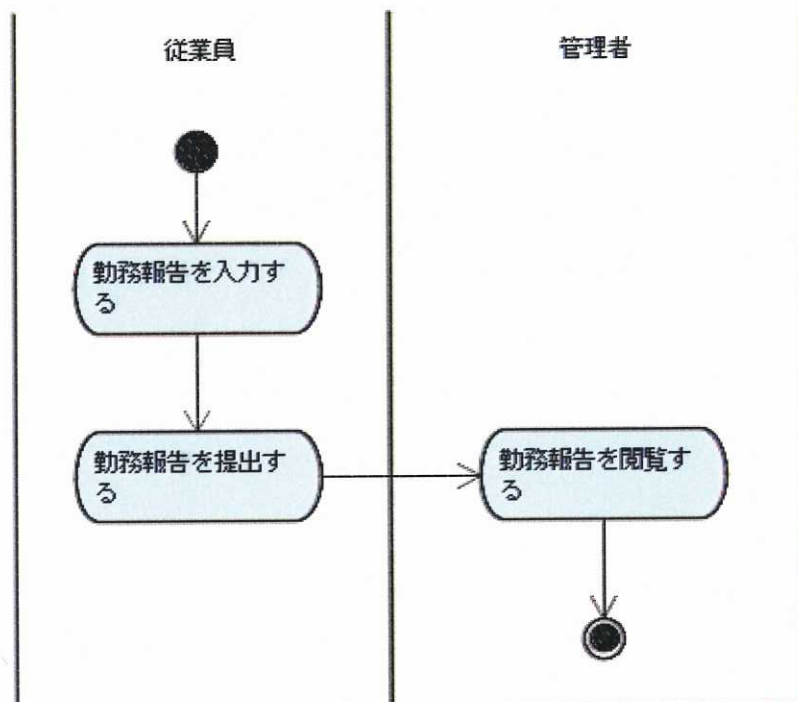
- 機能要件
 - 必須要件
 - ・ 従業員がクライアントに週次勤務報告を入力する際、既に月次勤務報告に
入力済みの勤務実績に関して、二重入力をなくすこと。
 - ・ 個人情報保護の観点から、クライアントアプリが週次勤務報告をファイル
として作成する際、データには個人の氏名を含まないこと。
 - ・ 管理者の設定（社員の誰がエリア管理者で、その管理者が誰を管理するの
かといった設定）を Web アプリ上で行えること。
 - 要望
 - ・ 週次勤務報告の時期に関して、やむなく定時に報告できない状況も想定し、
先行提出や遅れての提出を受け取れるようにしたい。
 - ・ 勤務報告の管理は印刷した書類上でも行われるため、印刷後も読みやすい
レイアウトで勤務報告を表示できるようにしたい。
 - ・ 通常の指揮系統別一覧のほか、エリア単位でも勤務状態を一覧表示できる
ようにしたい。営業担当者などが使用する。
 - ・ 既に提出済みの勤務報告を修正できるようにしたい。
- 機能外要件
 - 必須要件
 - ・ このシステムを導入することで、一部のシステム関係者に負担が一極化し
ないこと。
 - ・ A 社には社外常駐の従業員の割合が多いため、社外（イントラネットに接
続できない環境）でも利用できるシステムであること（A 社では社外向け
のサーバがないため、ブラウザでの Web アプリケーションの利用は社内
のみに限定される）。
 - ・ クライアントに特殊な実行環境（.Net フレームワークや JDK など）を必
要としないこと。MS・Office はすでに導入されていることを前提としてよ
い。
 - 要望
 - ・ 格納されている勤務実績は、最低 1 年間は DB に保存したい。

• ユースケース

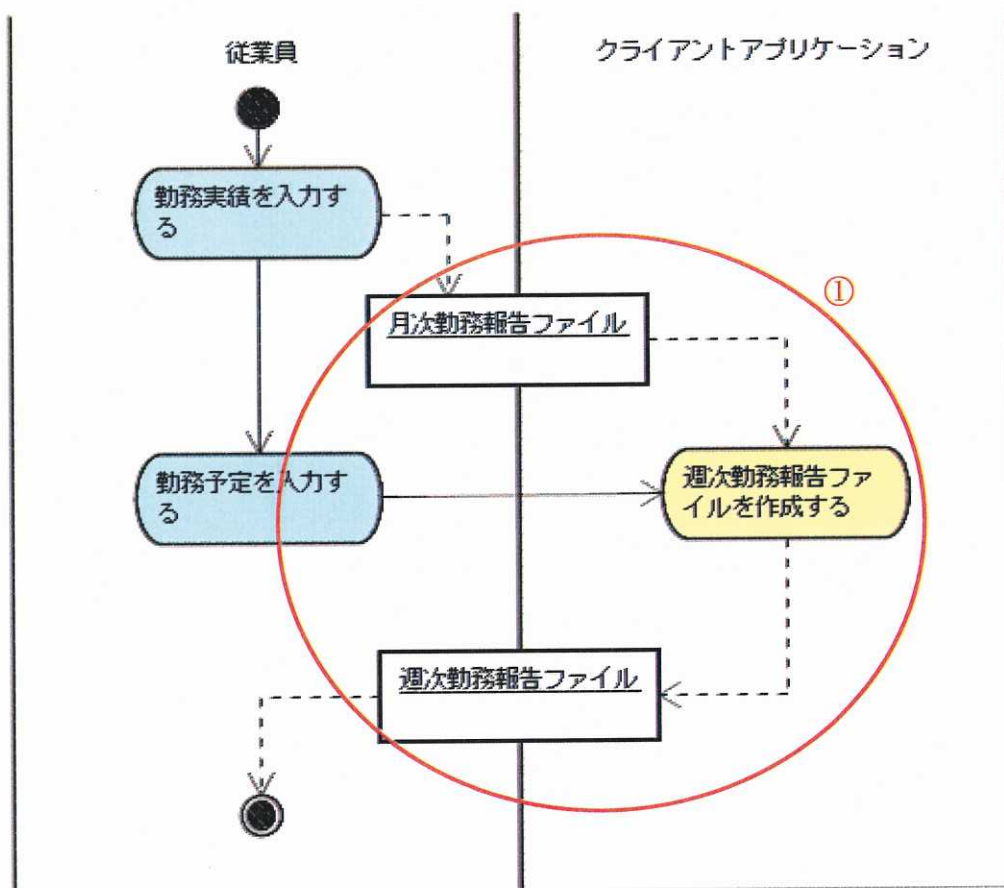


- 新運用フロー

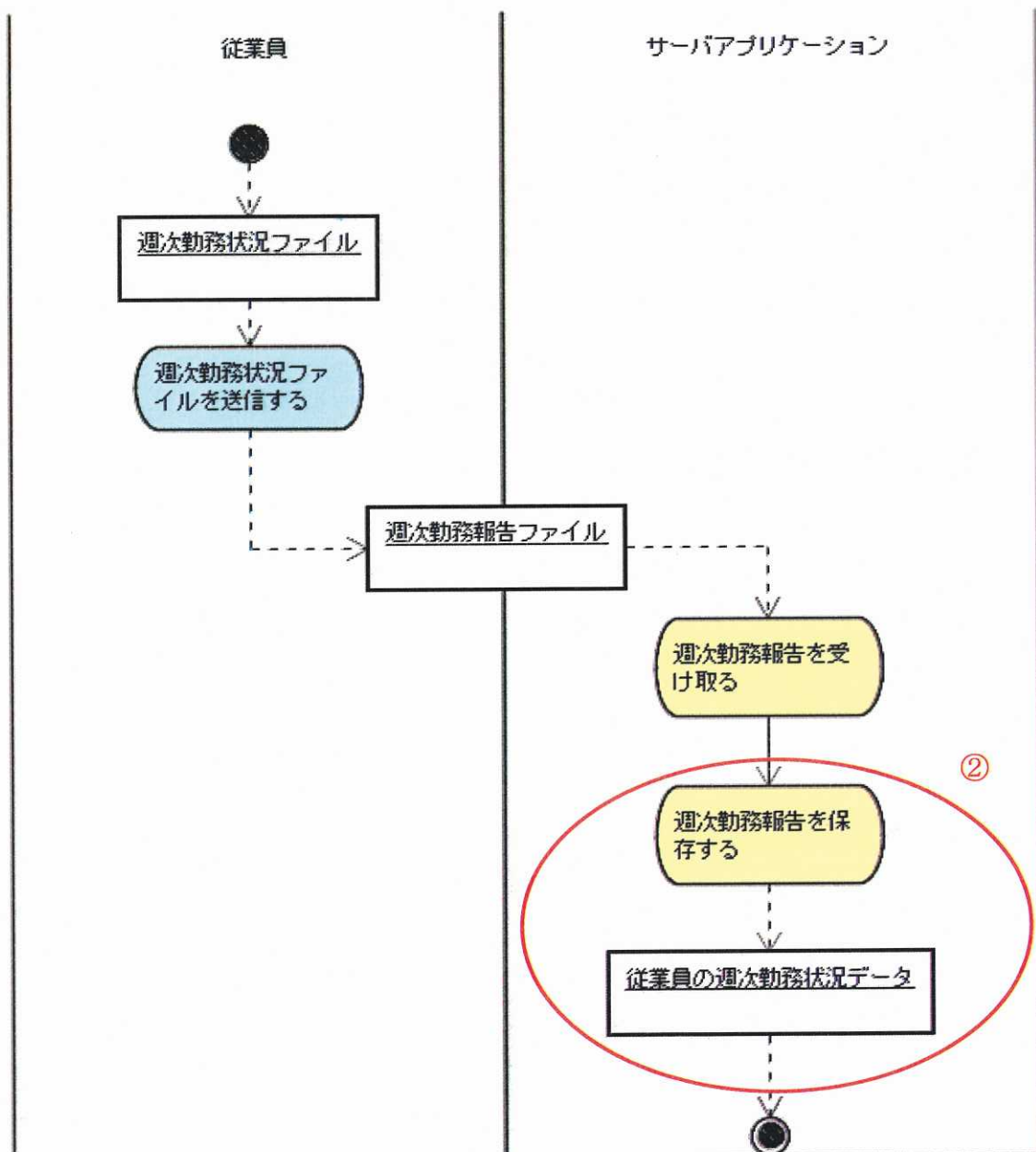
全体概念フロー



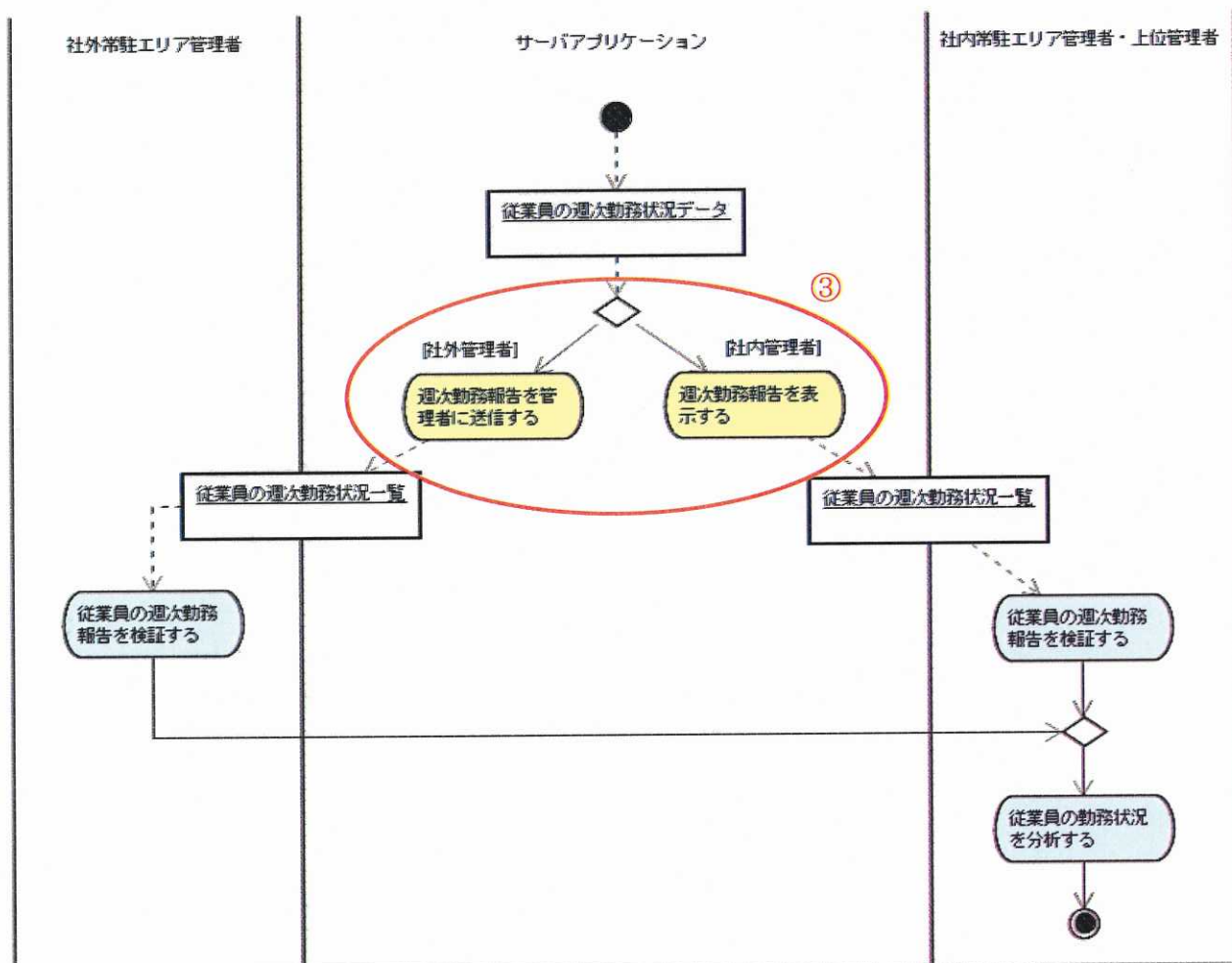
勤務報告を入力する



勤務報告を提出する



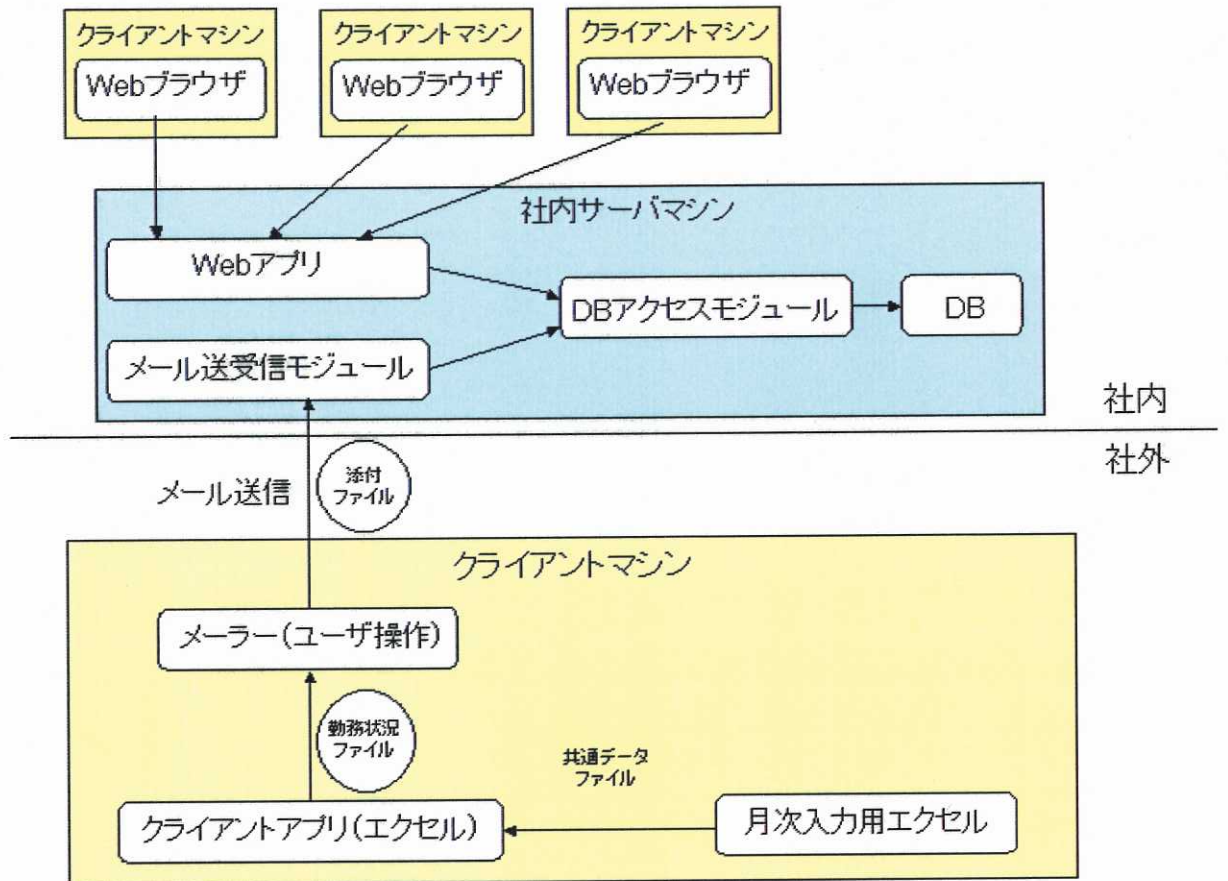
勤務報告を閲覧する



[新運用フローによる改善点]

- ① クライアントアプリケーションが、従業員が入力した月次勤務状況ファイルから自動的に勤務実績のデータを抽出してくることで、問題点 1 の二重入力を解消する。
- ② 従業員から送られた勤務報告を保存しておくことで、問題点 4 のデータの分散を解消する
- ③ 管理者の管理する従業員の週次勤務報告を一覧で表示・配信することで、問題点 2 の一つ一つ開く手間、問題点 3 の一覧で閲覧できない問題を解消する。

- システム構成



4.設計・実装

4.1. Web アプリケーション

Web アプリケーションの分析・設計では,

- ・ 画面遷移図
- ・ 画面モックアップ
- ・ システム構成図
- ・ ロバストネス図

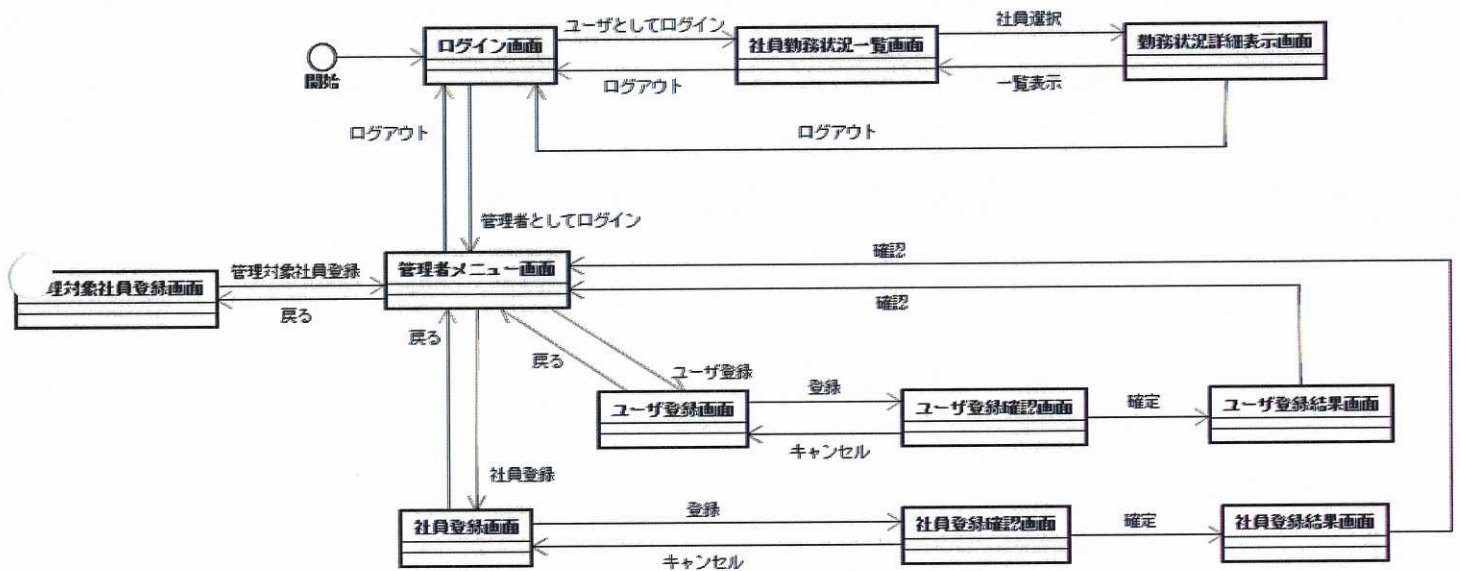
を作成した.

以下にそれらの図を掲載する.

4.1.1. 画面遷移図

下の図は、Web アプリケーションの画面遷移図である。

Web画面遷移図



4.1.2. 画面モックアップ

- ・ ログイン画面

下の図は，ログイン画面のモックアップである．

勤怠管理システム

ユーザログイン	管理者としてログイン
ID <input type="text"/>	ID <input type="text"/>
Password <input type="password"/>	Password <input type="password"/>
<input type="button" value="ログイン"/>	<input type="button" value="ログイン"/>

ログイン画面では，ユーザログインと管理者ログインが分かれている．

ユーザログインは，管理者が自分の配下の社員の勤務状況を閲覧するときに使う．

管理者としてログインの方は，新たに Web アプリケーションのユーザを登録したり，社員を登録したり，ユーザが管理する社員を追加したりする場合に使う．

- 勤務状況一覧表示画面

下の図は、勤務状況一覧画面のモックアップである.

社員勤務状況一覧

山田 太郎さん, こんにちは.

2006 ▼ 年 1 ▼ 月 3 ▼ 週 表示

氏名	項目	2日	3日	4日	5日	6日	7日	8日
石井 博史	実働時間(分)	0	450	750	540	420	0	0
	残業実績(分)	0	30	270	0	0	0	0
	残業予定(分)	60	0	0	90	0	300	0
	過不足(分)	-60	30	270	-90	0	-300	0
玉田 信弘	実働時間(分)	0	450	750	540	420	0	0
	残業実績(分)	0	30	270	0	0	0	0
	残業予定(分)	60	0	0	90	0	300	0
	過不足(分)	-60	30	270	-90	0	-300	0

[ログアウト](#)

勤務状況一覧画面では、自分の配下の社員の勤務状況を一覧表示することができる。閲覧したい年・月・週を選択し、表示ボタンを押すと、該当する時期の一覧が表示される。さらに詳しい情報を個別に閲覧したいときは、社員の氏名をクリックすることで詳細をみることができる。

- 勤務状況詳細表示画面

下の図は、勤務状況詳細表示画面のモックアップである。

社員勤務状況詳細

2006 年 1 月 3 週 表示

石井 博史さんの第 3 週の勤務状況

予定 Aプロジェクト 1次フェーズ コーディング Aプロジェクト 2次フェーズ 仕様検討 社内会議 中間面談

	2日	3日	4日	5日	6日	7日	8日
出勤時刻	-	10:00	9:00	8:00	10:00	-	-
退社時刻	-	17:30	21:30	17:00	17:00	-	-
休憩時間(分)	0	60	60	120	60	0	0
実働時間(分)	0	450	750	540	420	0	0
残業時間(分)	0	30	270	0	0	0	0
残業予定(分)	60	0	0	90	0	300	0
過不足(分)	-60	30	270	-90	0	-300	0
進捗状況	-	進	進	進	普	-	-
特記事項	-	午前休	午前休	休日出勤	ノ残業デー	休日出勤	休日休暇等
作業実績・問題点・反省点・残業理由など		"遅刻"	"遅刻分の穴埋めなんたらかんたら"	"早退"	"休日出勤"		

[一覧表示へ](#)
[ログアウト](#)

勤務状況詳細画面では、選択した社員の勤務状況の詳細を閲覧することができる。閲覧したい年・月・週を選択し、表示ボタンを押すと、該当する時期の一覧が表示される。全員の勤務状況を一覧表示したい場合は、一覧表示へをクリックすることで見る事ができる。

- ・ 管理者メニュー画面

下の図は，管理者メニュー画面のモックアップである．

管理者メニュー

山田 太郎さん, こんにちは.

[ユーザ登録](#)

[社員登録](#)

[ユーザの管理対象社員登録](#)

[ログアウト](#)

管理者メニュー画面では，ユーザ登録，社員登録，ユーザの管理対象社員登録を選択することができる．

- ・ ユーザ登録画面

下の図は，ユーザ登録画面のモックアップである．

ユーザ登録



The mockup shows a vertical form with the following elements:

- 氏名 (Name): A text input field.
- 社員CD (Employee ID): A text input field.
- メールアドレス (Email Address): A text input field.
- ログインID (Login ID): A text input field.
- パスワード (Password): A text input field.
- システム管理権限 (System Management Authority): A dropdown menu with "なし" (None) selected.
- Buttons: Two buttons at the bottom, "登録" (Register) and "戻る" (Back).

ユーザ登録画面では，氏名，社員 CD，メールアドレス，ログイン ID，パスワードを記入し，管理者権限の有無を選択して登録ボタンを押すことで新しいユーザを登録することができる．登録したユーザのログイン ID とパスワードを入力すれば，ユーザログインができるようになり，管理者権限を付けたユーザであれば，管理者としてログインすることもできるようになる．

- ・ 社員登録画面

下の図は、社員登録画面のモックアップである。

社員登録

氏名

所属

社員CD

プロジェクト名

プロジェクトCD

登録

戻る

社員登録画面では、氏名、所属、社員 CD、プロジェクト名、プロジェクト CD を入力することで、新しい社員を登録することができる。社員を登録することで、その社員の勤務状況を Web アプリケーションで管理できるようになる。

- ・ 管理対象社員登録画面

下の図は，管理対象社員登録画面のモックアップである．

管理対象社員登録

管理者CD

管理対象社員CD

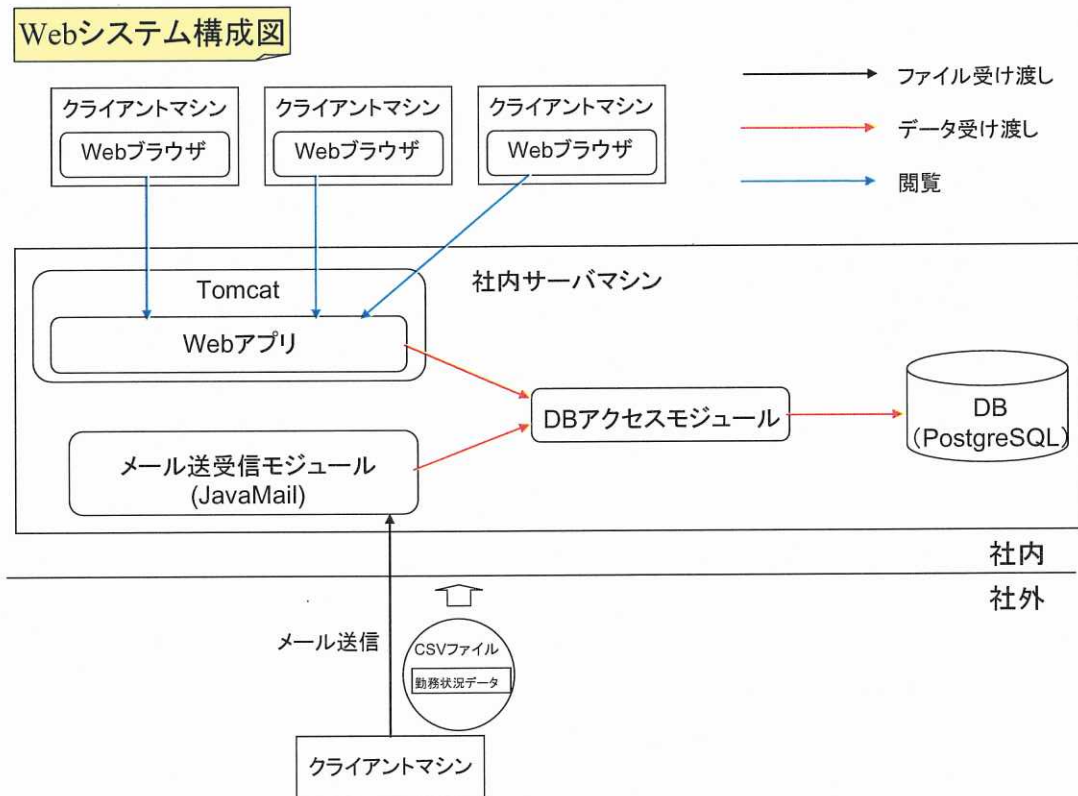
登録

戻る

管理対象社員登録画面では，管理者 CD と管理対象社員 CD を入力して登録ボタンを押すことで，新しく管理対象として追加ことができる．以降，管理対象として追加された社員の勤務状況は，管理者が閲覧することができるようになる．

4.1.3. システム構成図

下の図は、Web アプリケーションのシステム構成図である。

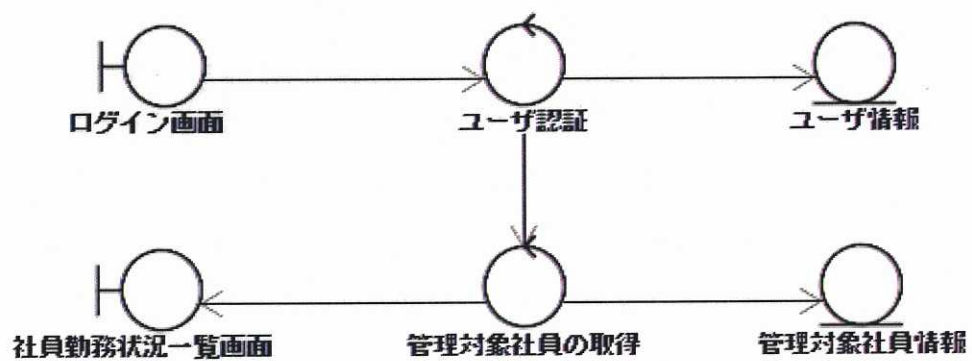


4.1.4. ロバストネス図

- ログインする

下の図は、ユーザがログインする際の動きについてロバストネス分析を行ったものである。

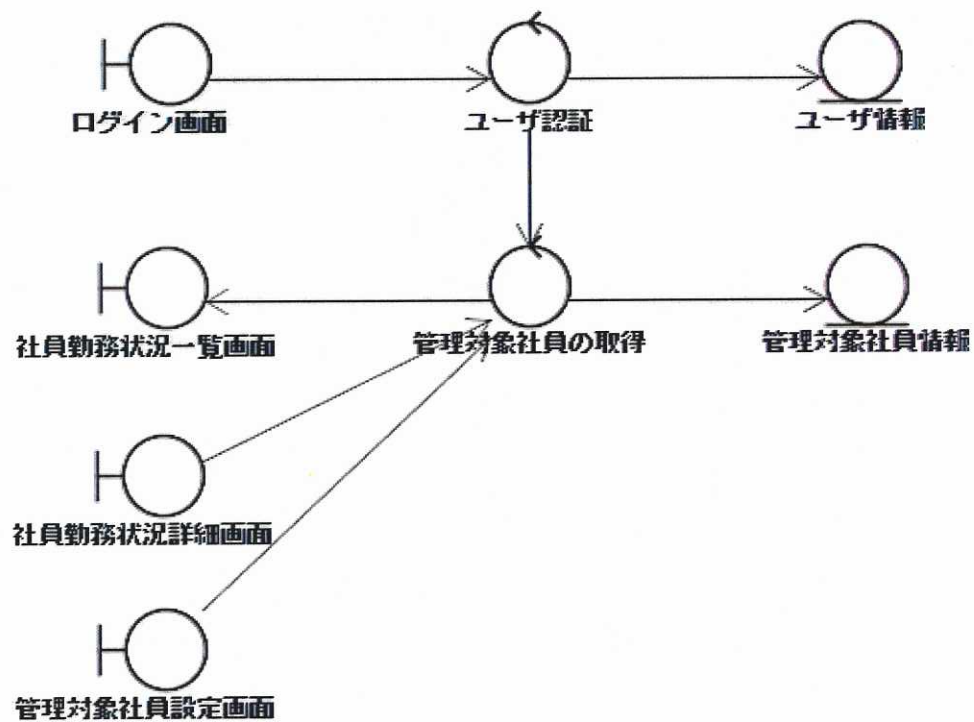
ユースケース：ログインする



- ・ 従業員の勤務報告を一覧表示する

下の図は、従業員の勤務報告を一覧表示する際の動きについてロバストネス分析を行ったものである。

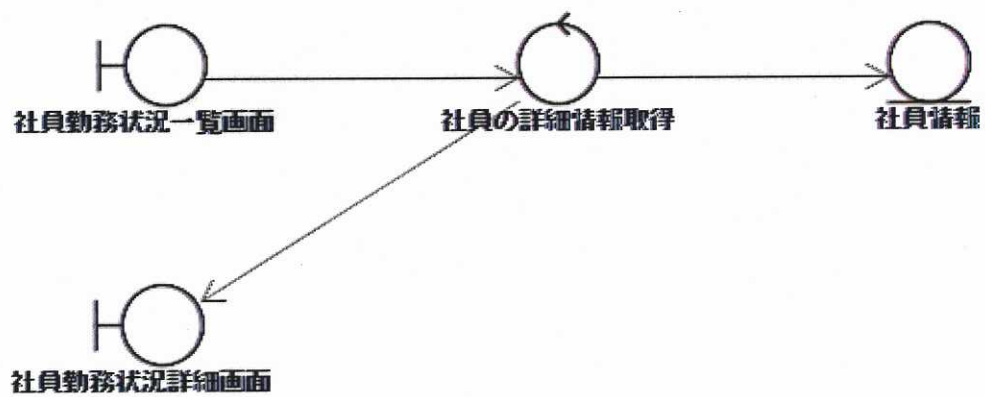
ユースケース：従業員の
勤務報告を一覧表示する



- ・ 従業員の勤務報告を詳細表示する

下の図は、従業員の勤務報告を詳細表示する際の動きについてロバストネス分析を行ったものである。

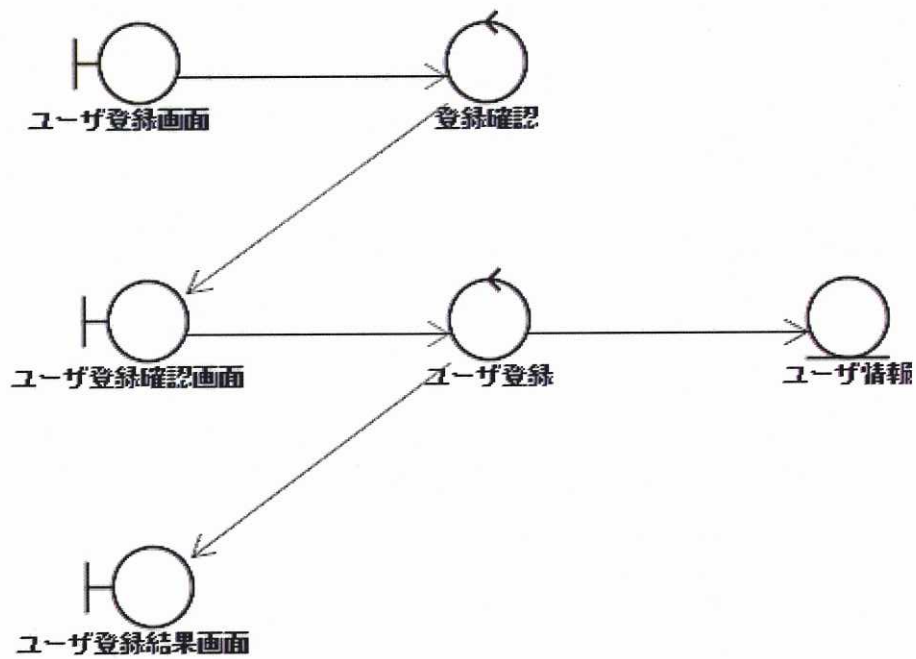
ユースケース：従業員の
勤務報告を詳細表示する



- ・ ユーザを登録する

下の図は、ユーザを登録する際の動きについてロバストネス分析を行ったものである。

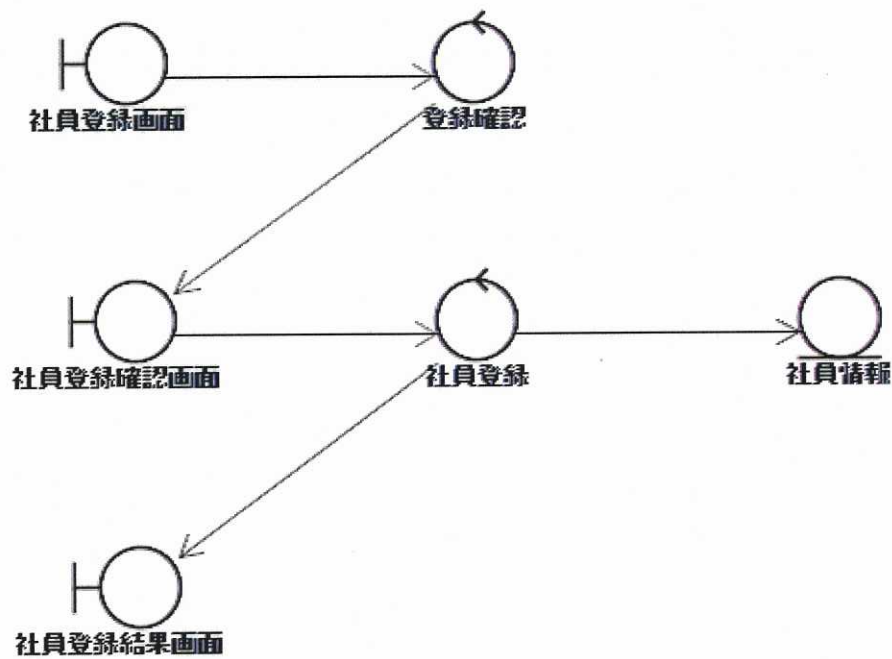
ユースケース：ユーザ登録をする



- ・ 社員登録をする

下の図は、社員登録をする際の動きについてロバストネス分析を行ったものである。

ユースケース：社員登録
をする



- ・ 管理対象社員を設定する

下の図は、管理対象社員を設定する際の動きについてロバストネス分析を行ったものである。

ユースケース：管理対象社員を設定する



4.2. クライアントアプリケーション

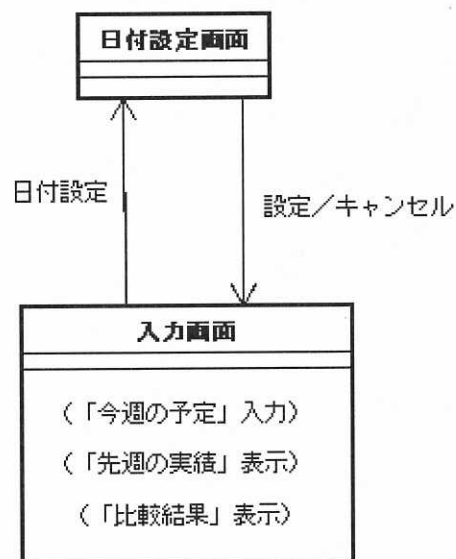
4.2.1. 概要

各従業員は、先週分の予定と実績の比較を行い、加えて今週分の勤務予定を立て、今週の勤務報告ファイルとして出力してメールで送信する。

主にこの勤務報告ファイルが含む内容は、先週に立てた勤務予定とそれに対する勤務実績、また今週分の勤務予定である。このうち先週の勤務予定と勤務実績のデータを、それぞれ先週の報告と、月次報告のデータから抽出することで二重入力を回避することができる。

このアプリケーションは、先週分の予定と実績のデータを自動で入力し、今週分の週次の勤務報告ファイルを作成するものである。便宜上、これをクライアントアプリと呼ぶ。

4.2.2. 画面遷移



クライアントアプリは基本的に入力画面のみであり、日付を手動で設定する場合のみ、日付設定画面を呼び出して入力する年月と週を設定する。

4.2.3. モックアップ

週間報告書		2 月度 第 2 週		所属CD: 123456		部門名: 事業部			
		平成18年1月23日 平成18年1月29日		社員番号: 789		氏名: 山田太郎			
今週の予定		日付	23日(月)	24日(火)	25日(水)	26日(木)	27日(金)	28日(土)	29日(日)
		特記事項	遅刻早退		遅刻早退		ノ残業デー	休日出勤	休日出勤
		残業予定	0:00	0:00	0:00	0:00	0:00	0:00	0:00
		Aプロジェクト 1次フェーズ コーディング	-	+	+	+	+	+	-
		社内会議	+	-	-	+	-	-	-
		中間面談	-	-	+	-	-	-	-
			-	-	-	-	-	-	-
			-	-	-	-	-	-	-
			-	-	-	-	-	-	-

先週の実績		出勤時刻	退社時刻	休憩時間	実働時間	残業実績	残業予定	過不足	進捗状況	作業実績・問題点・反省点・残業理由など
	16日(月)	12:00	17:30	1:00	4:30	0:00	0:10	0:10	遅	午前休
	17日(火)	11:00	22:00	1:30	9:30	1:30	0:20	1:10	普	検査により遅刻
	18日(水)	1:00	17:30	2:00	14:30	6:30	5:00	1:30	普	
	19日(木)	9:00	17:30	1:00	7:30	0:00	0:40	0:40	遅	
	20日(金)	9:00	17:30	1:00	7:30	0:00	3:00	3:00	遅	
	21日(土)	6:05	23:59	2:30	15:24	15:24	9:30	5:54	普	休日出勤
	22日(日)	0:20	5:50	0:30	5:00	5:00	0:20	4:40	進	
	所定労働時間	8:00		合計	63:54	28:24	19:00	9:24		確認印

クライアントアプリの画面は上図になっている。

大まかに 2 つの項目に分かれ、上段は今週の勤務予定の記入欄であり、下段は先週の勤務実績と先週に報告した勤務予定、両者の比較検討の記入欄である。

上段の最上部には日付の表示欄、個人情報の表示欄が存在している。日付は、クライアントアプリを起動した日が含まれる週を自動で表示するが、起動後に手作業で設定することも出来る。この週の数え方は独特であり、月次報告が 21 日開始で 20 日締めで設定されているために、「21 日を含む週の月曜日」から、翌月の第 1 週が始まる仕組みになっている。

今週の予定を記入する欄では、特記事項と残業予定時間を選択し、また今週の予定とその該当日を記入する。

先週の予定と実績を記入する欄では、先週の実績（出勤・退社・休憩の時間）と先週報告した予定が自動で入力され、差分が自動で表示される。これを参照して分析を行って記入し、今週分の勤務報告ファイルとして作成する。

4.2.4. 出力データ

出力するファイルは CSV 形式を採用する。

◇ファイルの中身

-----ここから-----

```
999,2112,9,3,  
0,3,3,2,7,2,1,  
100,0,0,130,0,500,0,  
"Aプロジェクト 1次フェーズ コーディング",0,0,1,1,0,0,0,  
"Aプロジェクト 2次フェーズ 仕様検討",0,0,0,0,1,1,0,  
"社内会議",1,0,0,0,0,0,0,  
"中間面談",0,0,1,0,0,0,0,  
  
,,,,,,,,,  
  
,,,,,,,,,  
  
0,0,0,2,,  
0,0,0,2,,  
1000,1730,100,3,"遅刻",  
900,2130,100,3,"遅刻分の穴埋めのために残業",  
800,1700,200,3,"早退",  
1000,1700,100,2,"休日出勤",  
0,0,0,2,,
```

-----ここまで-----

- ・ 1行目：社員番号と年月週
- ・ 2行目：今週の特記事項の予定
(0 から番号順に。空白、休日休暇等、休日出勤、午前休、午後休、遅刻早退、シフト、ノー残業デーとなる)
- ・ 3行目：今週の残業予定 (330 なら 3:30。同様に、時間はすべてコロンを省略)
- ・ 4行～9行目：今週の活動予定と該当日 (1 ならその日に該当、0 は該当しない)
- ・ 10行～15行目：先週の勤務実績と評価
(順に、先週の出勤時間、退社時間、休憩時間、自己申告による進捗状況、備考欄)
(※ 進捗状況は 0 から 3 まで、順に、空白、遅、普、進)

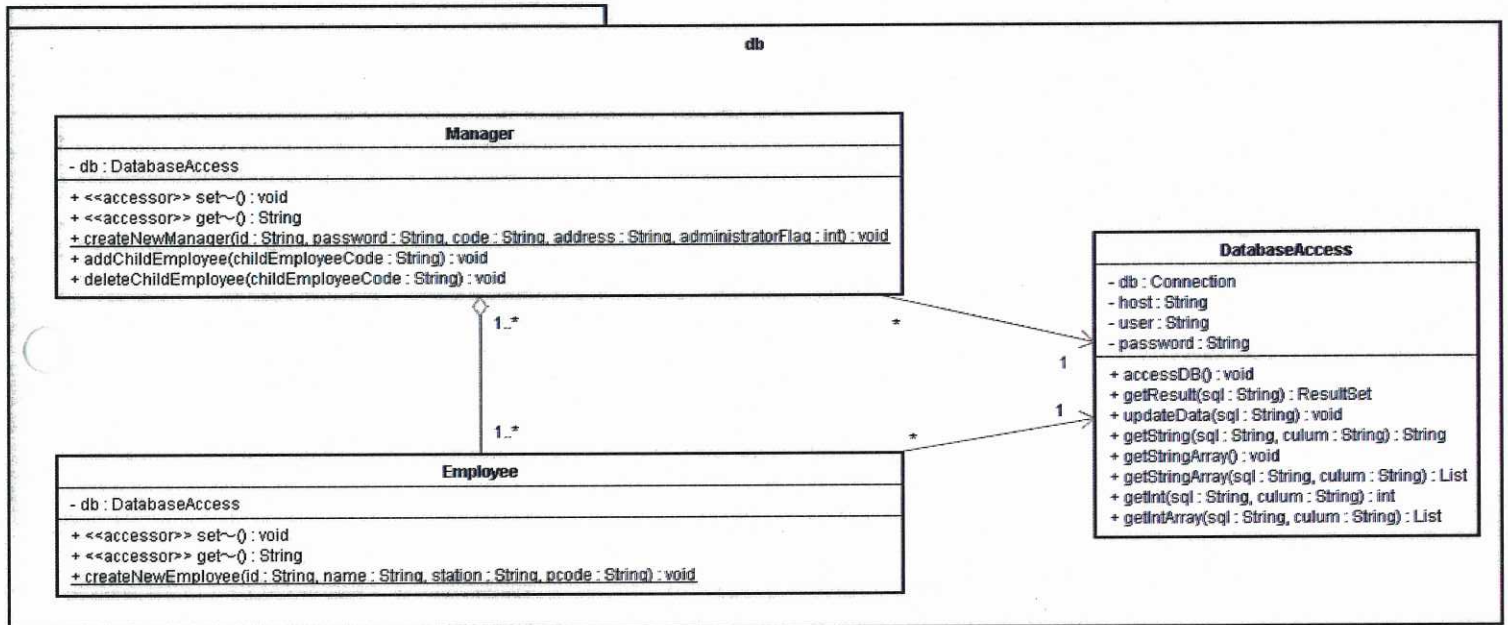
◇ファイル名 ex. 「0099921120903.next」

- ・ファイル名は、以上のような 13 桁のコード+拡張子になっている
- ・はじめの 5 桁は社員番号を 5 桁で表示したものである
- ・後半の 8 桁は、年と該当月と該当週
(月や週が一桁でも、02 や 06 のように用いる)
(上の例は 2112 年 9 月度の第 3 週)
- ・拡張子は、「.next」とする

4.3. メール送受信アプリケーション

4.3.1. クラス図

データベースアクセス部



・データベースからデータを持ってくる際に利用するモデルクラスとデータベースにアクセスするクラスからなる db パッケージ

・Web アプリケーション、メール受信アプリケーション、メール送信アプリケーション内で利用される

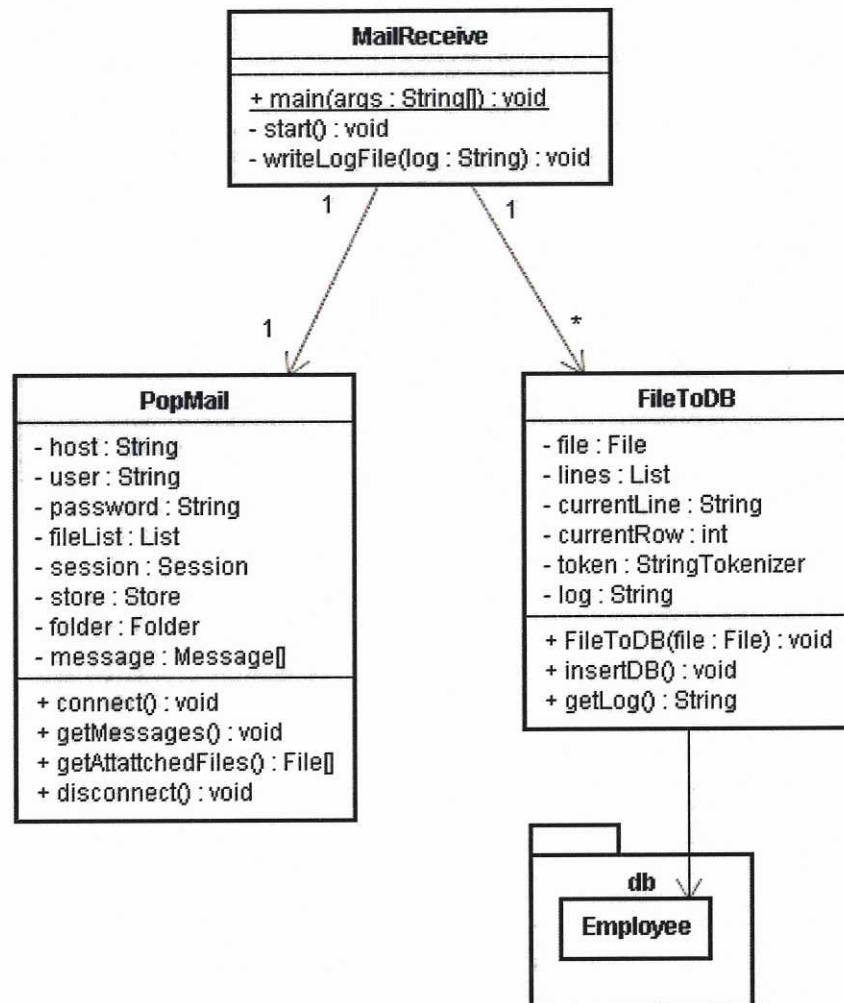
・Manager クラスと Employee クラスは、それぞれ一人の管理者・一人の従業員の情報をもつクラスである

・Manager クラスと Employee クラスのアクセサとして実装される `get~`, `set~` メソッドでそれぞれ管理者、従業員のデータを取得・格納する

・取得できるデータは、管理者クラスは『ユーザ ID』『社員番号』『名前』『システム管理者権限』『メールアドレス』『管理対象とする従業員』、従業員クラスは『社員番号』『名前』『所属』『プロジェクト番号』『プロジェクト名』『勤務実績情報』『勤務予定情報』

・管理者は一人以上の管理対象とする従業員をもち、従業員は一人以上の管理者に管理されるため、多重度は双方とも 1..*

メール受信部



・メールを受信し、取得した勤務報告ファイルからデータを読み込みデータベースに格納するアプリケーション

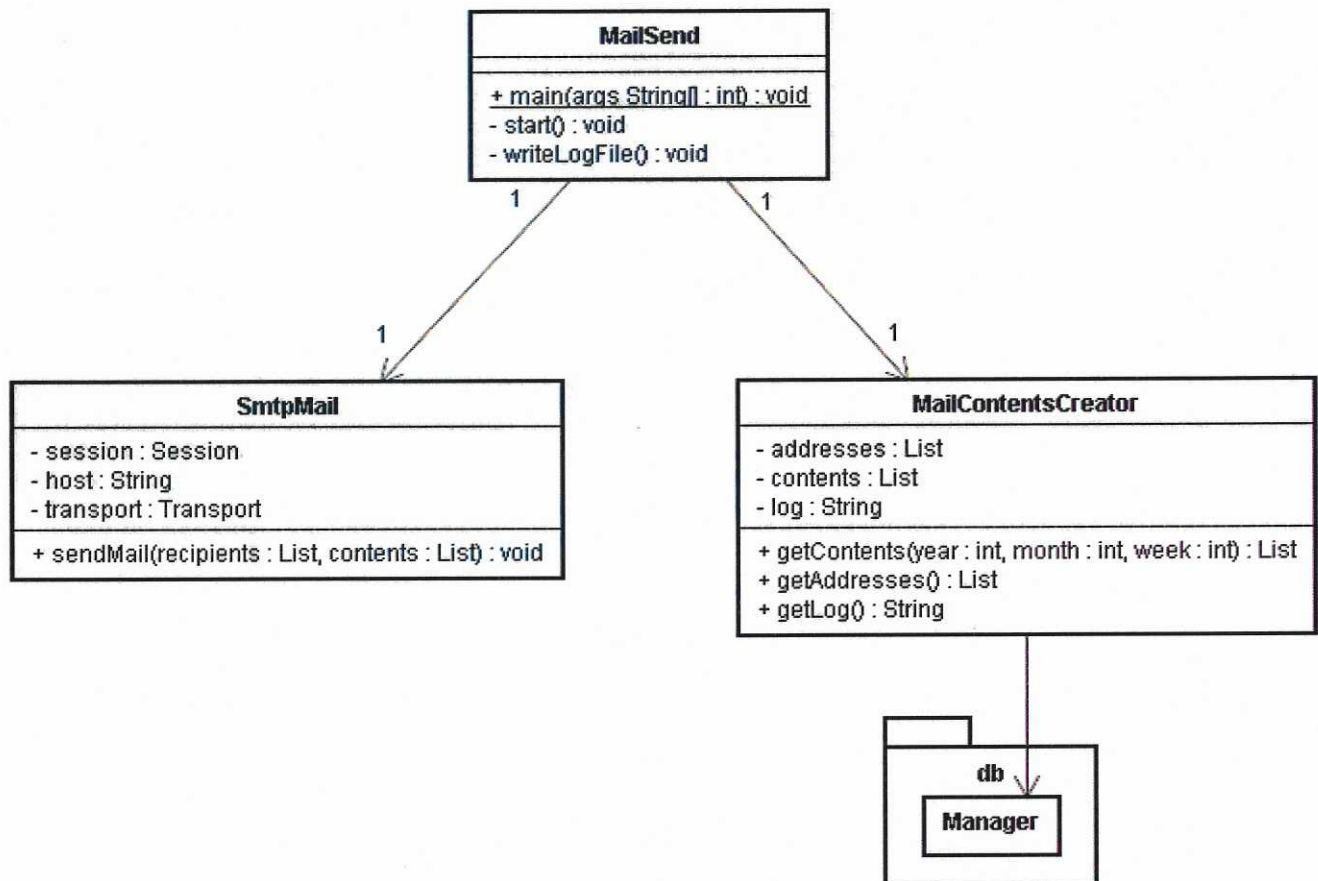
・MailReceive クラスは、実際にアプリケーションを実行する main 関数を含んだクラス

・PopMail クラスは、POP サーバにアクセスしメールを受信し、添付ファイルを取得するクラス

・FileToDB クラスは、渡されたファイルを解析して、ファイルが含むデータをデータベースに格納するクラス。格納時には db パッケージの Employee クラスの set~メソッドを利用

・main 関数が実行された MailReceive クラスは PopMail クラスでメールを受信、添付ファイルを取得後、FileToDB クラスに添付ファイルを渡してデータベースに勤務報告データを格納する

メール送信アプリケーション



- ・データベースから勤務報告データを取得し、管理者に勤務報告メールを送信するアプリケーション
- ・MailSend クラスは、実際にアプリケーションを実行する main 関数を含んだクラス
- ・SmtplibMail クラスは、SMTP サーバにアクセスしてメールを送信するクラス
- ・MailContentsCreator クラスは、データベースから勤務報告データを取得し、メール本文に記載する文章を作成するクラス。データベースからデータを取得する際、db パッケージの Manager クラスの get~メソッドを利用する
- ・main 関数を実行された MailSend クラスは、MailContentsCreator クラスで送信するメール本文と宛先を取得し、SmtplibMail クラスでメールを送信する

4.3.2. 送信されるメール本文のテンプレート

田中 太郎(10225)

月： 9:00 - 18:30(+0:30) 計 8:30 月： 1:00

火： 9:00 - 19:00(+0:00) 計 9:00 火： 0:30

水： 9:00 - 20:00(+1:00) 計 10:00 水： 1:00

木： 9:00 - 18:00(-0:30) 計 9:00 木： 1:00

金： 9:00 - 21:00(+1:00) 計 11:00 金： 1:30

土： -----(-) 計 ----- 土： -----

日： -----(-) 計 ----- 日： -----

- ・ 1行目は名前と社員番号
- ・ 2行目以下は勤務報告
- ・ 数字は左から、『出勤時間』『退勤時間』『予定された残業時間との過不足』『合計勤務時間』『次週の残業予定時間』
- ・ 『予定された残業時間との過不足』は先週報告された『次週の残業予定時間』と今週の報告に含まれる残業時間の過不足である
- ・ 『合計勤務時間』は『出勤時間』と『退勤時間』から導き出される合計勤務時間から休憩時間を引いた時間である

5. 個人レポート

5.1. 藤原育実

私が今回参加したプロジェクトは、社会人の方が PM となり、実際に会社で発生している問題を解決できるソフトウェアを作るということを目的としていた。今回のプロジェクトでは、ソフトウェアを使っていただく顧客がいて、顧客の納得するものを作らなくてはならないという点で、従来のプロジェクトとは異なっていた。そのため、自分たちの好きなように作りたいものを作っていればよいというわけではなく、相手の業務や制約を分析し、制約を守りつつ問題点を解決できるシステムを提案しなくてはならなかった。私は今までこのような経験をしたことがなかったため、悩んだことや、苦労したことも多かったが、学んだことも多くあった。

5.1.1. 業務分析について

まずは、業務分析について学んだことについて述べる。業務分析では、状況を性格に把握し、問題点を明確にするということの重要性を学んだ。我々が今回改善を試みたのは、会社の勤務状況管理であるが、会社の従来の勤務状況管理は独自の用語や規則を用いて規定されていたので、用語や規則について詳細に理解する必要があった。これらの用語や規則についての理解が曖昧なまま進めようとする、必ず後で誤解や混乱が生じてしまう。そのため、会社にヒアリングを行ったり、従来使われていた勤務状況管理のためのファイルを見せてもらったりしながら、分析を行った。このような分析を行うことで、問題点や改善できそうな部分が明確になり、次のフェーズに進むことができるようになる。このように、状況を把握し、問題点を明確にするという分析の作業は、今回のように会社向けのソフトウェアを作るようなケースだけに限った話ではない。ソフトウェアは、対象となるユーザを想定し、ユーザが何らかの問題を解決できたり、利益を得られたりすることを目的に作られる。であるから、ユーザがどのような問題を抱えていて、どのような利益を求めているのかということ曖昧なままソフトウェアを作成しようとしても、有用なものを完成させるのは難しい。以前の私は、このような分析作業の重要性をあまり認識していなかったので、今回のプロジェクトでは、その重要性を学べたという意味で成長できたのではないと思う。

5.1.2. 要件定義について

次に、要件定義について学んだことについて述べる。要件定義では、業務分析で明らかになった問題点を解決するために、満たさなければならない要件を決めていく。要件定義は、業務分析がしっかりとできていないと進めることができない。たとえば、相手の会社では、月次報告と週次報告のエクセルファイルを用意し、入力しているため、二重入力

手間が発生してしまうという問題と、管理者が全員分のエクセルファイルを管理するのが大変であるという問題が発生していた。そこで私がすぐに思いついたのは、エクセルファイルをすべて Web アプリケーションに置き換えるという方法であった。しかし、相手の会社では月次報告や週次報告を記入する社員が、Web を利用できるような環境にいない可能性があるという問題や、そもそも相手の会社に外部に公開しているサーバがないという問題があったため、この案は却下された。もしも相手の会社の抱える制約を把握しないまま要件定義を行おうとしてしまったら、単純に Web アプリケーションに置き換えるという手段を提案して、却下されてしまったかもしれない。要件定義は業務分析に基づいて行わなければならないということを再認識することができた。また、要件はすべてこちら側から相手に提案し、それに対して意見や承認をもらうという方法を採用しなければならないということも学んだ。こちら側から提案せず、相手側から出された要求をひたすら飲むという方法を採用していると、相手側はこちら側の抱える条件や制約を無視して提案をしてくるため、要件が膨大に膨らんでしまい、実現不可能になってしまう可能性が高い。であるから、受け身で相手の要求を飲むだけではなく、相手が納得できるような要件をこちらで定義し、提案していくことの重要性を学ぶことができた。

5.1.3. 設計・実装について

最後に、設計・実装について学んだことについて述べる。設計・実装フェーズで私が担当したのは、Web アプリケーション部分である。Web アプリケーションを一から設計・実装するのは私にとって初めての経験だったので、苦労した点も多かったが、学んだ部分も大きかった。設計では、画面遷移図やロバストネス図を作成した。これらの図によって、Web アプリケーションにはどのような画面があって、どのような機能があれば要件を満たせるのかということを明確にしていっていった。しかし、実際に設計してみると、足りないものや、不十分なものが多数出てきて、何回も改善することになった。設計でも、要件定義フェーズで決まったことをきちんと確認しながら進めていかなくてはならないということを実感した。実装では、struts フレームワークを用いて、JSP/Servlet で実装した。Struts や JSP/Servlet を用いて一から自分でアプリケーションを実装した経験がなかったため、技術調査を行いながら実装を進めていかなくてはならなかった。試行錯誤しながら実装を進めていくと、それらの仕組みや長所・短所がよくわかるようになる。大変な作業ではあったが、勉強になり、今回学んだことは今後 Web アプリケーションを作成する際にも活かしていけると思う。

以上が、今回のプロジェクトで学んだことである。今回は主に実装以前の段階の重要性というものを強く認識させられた。私はこれまで実装重視で、それ以前のフェーズは軽視する傾向にあったため、今回のプロジェクトでは苦痛だった点も多かったが、目から鱗が落ち、ソフトウェア作成に対する考え方が変わったところも多くある。今回学んだことを、

今後のソフトウェア開発にもいかしていきたいと思う。

レポート課題名：「研究成果」

授業科目：研究プロジェクト2

担当者名：大岩元 先生

環境情報学部3年 野上大輔(70347866/t03786dn)

<はじめに>

この研究室との出会いは予想外であり、オブジェクト・プログラミングの履修者としてメールを頂き、それに誘われる形で研究室に入室する運びとなった。

開始当初は、ソフト開発の面ばかりに興味を示しており、単なるインターンシップの延長のような観念で見えていた。この夏にインターンシップを体験したばかりであり、その時は課題を与えられてその指示通りの開発を行うという内容だったため、この研究プロジェクトも同様に思っていた。

この認識が、すぐにも間違いだったことが判明する。それは初めての経験が豊富に盛り込まれていて、いい意味で裏切られたと表現してもよい。

<精神面>

「経験」は、この研究会のスタートのときに自分が宣言した言葉である。様々なものを見て、様々なことを聞きたい、と最初の抱負を語った覚えがある。

すべてが片付く万能な言葉で、簡単にその場をやり過ごそうとしていたようにも見えるが、実際に求めていたものはこれ以上でもこれ以下でもない、本当に「経験」の一言でしか片付けられないことだとも思う。

さて、いざプロジェクトが組織され、すぐに開発に着手すると思っていたのだが、設定されたスケジュールを見る限りでは開発は後半の僅かな時間に行われる予定となっていた。開発にあまり時間を割かなくて良いのかという不安もあったが、開発までに何をして過ごすのかという疑問が最も大きかった覚えがある。

正直、プログラミング、つまりは開発に誘われて入室したために、このスケジュールには実のところ不満を覚えた。始めの行動として業務の分析から着手することになっていたが、当時は分析の有益性が理解できず、分析や調査は他のメンバーに任せて自分はその情報をあとで解説して貰おうなどと甘い考えを抱くこともあったのも否めない。

しかし全員の足並みの揃わなくてはプロジェクトが前進せず、少なくとも進行を滞らせる結果となったのは明白だった。認識の落差が最も致命的な問題であり、その落差を解消するために多くの時間を有してしまったことが、今思えば最後まで響いてしまったようにも思う。

今思うと、本来ならば分析という作業に頭を悩ませるのは、当然ながら初めての経験で

あり、さぞや発見に満ちた機会だったことだろう。しかし、なにぶん時間の制限がきつく、初めての挑戦の割には片手間になってしまったことが残念である。

分析の最終段階として、企業へ直接ヒアリングできたことが、何よりも貴重な経験だった。その準備に追われ、寝ても醒めても研究会の課題のことを考えて過ごしていた時期もあったが、これが実際のコンサルティング業務かと思うと決して辛いばかりではない時期だった。

そうして準備に準備を重ね、いざ何うまでは自信に満ち満ちていた内容にも拘らず、即座にいかに思い上がりだったかを知らされてしまう。確かに自分たちよりもシステムに触れる機会も多く、長所短所も把握していることもあったろうが、それでも自分たちの見通しが甘すぎたのは紛れもない事実だったと思う。

なあなあのまま、厳密に意味も考えずにお茶を濁そうとしていた内容が多く、悉くその甘さを指摘されたのは当然のことだったろう。だが少なくとも、その指摘の甲斐あって、全員の中に適当に誤魔化すのは許されないという認識も広がり、いい刺激になっていたと確信している。

だが一方で、現場の視線というものに感嘆しつつも、自分もそうしたレベルになれるのかという不安も実は否めなかった。こればかりは場数を踏む以外に王道はないだろうが、漠然とした一抹の不安を覚えてしまうのも無理からぬ程、ヒアリングという機会は衝撃だったと感じている。

毎週木曜の報告会では、プレゼンテーションの機会も幾度となくいただけたが、これに関しては目に見えての上達の兆しが現れずに残念である。緊張によって動悸が激しくなるなどの異変はないものの、思考回路は破綻しており、シミュレートしていた内容の半分も話せることはなかったのが現状である。

積極的に挑んだだけにこの低成長は残念であるが、改善のアドバイスを少しずつ頂けたことを、これまた少しずつ噛み締めていければと思う。

思えば SFC 入学から 2 年半で研究会に飛び込んだこととなるが、ほんとうに多くのものを経験した時期であった。経験していないものと言え、あとは残留程度のものである。

たった一度でも、この経験の一つ一つが自分の糧となるのは明白であり、心に留まっている格言だけでも枚挙に暇がない。

もちろん不満もなかった訳ではないが、ひとつの形として締め切りを迎えようとしている今、達成感と満足感の方が遥かに大きい。

<技術面>

開発技術の知識に関して疎かった自分は、業務分析から要件定義の段階で、「どの技術でなにが実現できるのか」が完全に解らず、こればかりは勉強不足を咎められても仕方がなかっただろう。このプロジェクトを通じて、新しい知識を多く得ることとなったのは願ってもない成長だった。

まずは開発以前の段階として、UMLの学習がある。これは同時履修していた「オブジェクト指向開発」の成果もあり、非常に丁寧に学習することが出来た。初めはその意図が解らず面倒に感じることも多かったが、開発が進むにつれて初期の分析がいかに必須かを思い知らされる機会が多かったために、非常に意義のある内容だった。

技術とは別の話かもしれないが、この開発に当たり、非常に多くのチャート図を書いた。HCPチャートの存在を教わった当初は、その意図も分からずにただただ面倒な作業としての認識しかなかったことがまるで嘘のようである。その頃は、おそらく複雑なプログラムを組む機会もなく、自分の思いつくがままの、いわば拡張的なプログラミングをしていたことだろう。しかし今回は対照的に、ある目的を厳密に成し遂げる、いわば収縮的なプログラミングを求められていたために、全体を常に監視しながらまとめるように実装に当たっていた、と考えている。

そのための俯瞰的な分析、それを実現するのに適切だったのがHCPチャートなどの可視化作業なのだろう。チャートなどを利用して可視化することがいかに大事か、身を持って体験することが出来たのも、思わぬ収穫だったと感じている。

何が学べるか分からないからこそ、未知への挑戦はやめられない。

他にもSQLやサーバプログラミングの基礎を学ぶことが出来た。なかなか独習では始めにくいこともあり、講習という機会を設けていただいて丁寧な学習が出来たことは非常にありがたかった。惜しむらくは、あまりそれらの学習成果を活用できなかったことにある。

実際の開発の段階では、3人で作業を分担して進行させた。自分は専らローカル環境下でのプログラムと格闘することになり、直接データベースアクセスやサーバレットに触れる機会をあまり取れなかったのだが、ややその点に関して残念さが否めない。

他の2人がネットワークで直接関与するに対して、自分の担当とは直接関与することはなく、アプリが作成するファイルを介しての連結だった。よってソースを見合わせることもなく、作成するファイルの形式だけを同調させて、後は完全に個別の作業になっていた。

時間が押していただけに、単独での作業は障害もなく進行できて救われたとも言えるが、情報を示し合わせて連結を行う共同開発にももっと関与したかったのも事実である。

そうしてせっかく土曜日に開講していただいた講習会で習った、JSPやSQLなどを実装する機会がほとんどなく、一方でVBAという予想外の言語を扱うこととなり全く当初から

は予想も付かない展開だった。自分は唯一 Visual Basic の経験があったため、この分担は適任だったと思うし、何よりも自分で志願した結果でもあった。

本音を言えば、願わくは、全ての開発に満遍なく関与したかったのだが、こればかりは時間の都合やプロジェクトという特性上は致し方ないかもしれない。

先学期はオブジェクト・プログラミングの講義でペアでの開発に当たり、初め個別に作業を行いつつ、最終的には連結する計画を立案していた。

しかし、オブジェクト指向ならば分担が容易と考えていただけに、連結に予想外に手間取ったことはいくらか衝撃であり、苦いがよい経験だった。

今回は前述のように分担の性質上、連結面の苦労がほとんど無くこうした苦労に直面しなかった。楽には越したことは無いが、やや難を言うならば、グループによる作業という実感が今ひとつ沸きにくかったことがある。

<おわりに>

去る 1 月 30 日に実際に来社して、一足早く成果物のデモンストレーションを実施した。本プロジェクトの一つの集大成であり、思えばこの日のために 3 ヶ月を過ごしたといっても過言ではない日である。

とは言え、実のところ自分の担当箇所には改善の余地が多数含まれていた。その場では最低限の機能を実践して概要を説明するものだったとは言え、最終的な発表という場面で成果物として満足のいくものを披露できなかったのは残念なことだった。

その場では好印象を頂けていたものの、やはり当初の目標に到達できなかったこと、想像していたものを作り出すに至らなかったことは、残念以外の何物でもない。

全体を振り返ると、当初のスケジュールに比べて大幅な遅れが生じていたことが開発の遅れ、果てには完成の遅れに繋がったのは分析するまでもない。そして当初のスケジュールが達成できなかったのは、ヒアリングの日程がなかなか決定しなかったこともあるが、そのヒアリングに対しての準備、即ち分析段階に手間取ったことが一番の理由であろう。

初めての研究会であり、初めてのプロジェクトだったために致し方ないとも言いつけることは出来るが、出来ることならこの反省を活用して次回のプロジェクトに反映させたいと思う。

この研究プロジェクトのみならず、「オブジェクト指向開発」や「情報教育論」の成果も手伝って、貴重な経験を過ごせた半期だったのは間違いないと思う。

たくさんの人のお世話になり、非常に多くのことを見聞きできた。大いにお世話になった方々への感謝とともに、一つのプロジェクトの終わりを迎えようと思う。

本当にありがとうございました。

大岩研究会 2

コラボレイティブマネジメント

最終個人レポート

環境情報 2 年 70440347(t04034ra)

安藤 亮一

僕がまずこの研究会に参加したいと思ったきっかけは、「企業と連携したソフトウェア開発」というテーマに強く惹きつけられたからです。前々から、授業などでプログラムしたりソフトウェアをつくったりはしてきたのですが、実際に企業に入ったときに、ただコードを書くだけの今のやり方をそのままソフトウェア開発につかっていけるのかどうか、不安でした。それは実際に現場の空気に触れなければ絶対にわからない、大学でただ授業を受けているだけでは決して経験することはできないと思っていました。そのときにこの研究会のテーマを見て、「これなら現場の空気を少しでも体験できるのではないか」と思い、足を踏み入れました。半年プロジェクトを体験した結果としては、わずかながらその空気を体験することができたと思います。そもそも僕は今までグループによるソフトウェア開発というものを体験したことがなく、そのプロセスの”いろはのい”も理解していませんでした。そんな状況でいきなりプロジェクトグループに放り込まれて、要件定義・分析・設計と初めてのソフトウェア開発プロセスを次々と体験していった、それだけでも充分貴重な体験ができたと思います。ですが、なにより僕を刺激したのが、「実在企業への提案」でした。

最初に研究テーマを決めるときに、この「実在企業の勤務状況管理システム」を選択したのは、「実在の企業と関わるテーマならちょっとくらい企業におけるソフトウェア開発の空気に触れられるかな」とぐらいに考えていたのですが、実際にこのプロジェクトに参加することになって、その実在企業にこちらのシステム案を提案するという話に具体性が出てきたとき、初めての経験に若干興奮するのと同時に、不安を感じました。その提案の要になる要件定義なんていう開発プロセスを今までに体験したことなどもちろんなく、その進め方すら全く理解していなかったからです。

こんな状況でチームの最初の会議に入り、「さて、まず最初に自分達は何をすればいいのかをはっきりさせよう」となったのですが、こういった場合、ただ「何をすればいい？」といっているだけでは答えなどせず、だらだらと時間が過ぎていくのが今までのグループでの作業でした。ですが今回は、「何をすればいいかを明確にするために、まず自分にどんな問題があり、その問題を解決するためにはどうすればいいかを紙に書いていこう」という方法をとることになりました。実際にこの方法で自分の問題の What と How を並べていき、それをチーム全体で共有したとき、チーム全体の問題点とその解決法が現れて、それを解決することが現在自分たちがしなければならないことであるということがわかりました。こ

んなやり方で明確にグループで自分達がすべきことが決まったことは、僕にとっては本当に目からうろこであり、確実にこれから様々な場面で使っていける手法だと感じました。

その後、実際にその会議で明確になった「今回のプロジェクトに対してのチーム内での認識がばらばらである」という問題を解決するために、「実在企業の社員である佐々木さんにヒアリングするという形で企業の勤務状況管理制度の現状を理解する」という解決方法でこの問題を解決し、全員の認識を一致させるところから要件定義のプロセスに入っていました。そして、このプロセスの最終段階の作業として「実在企業への提案」を行うことになりました。

システム案の提案という形で企業へ赴き、現在こちらが考えているシステムを発表し、それに対する社員の方々の意見を伺うという流れになりました。ここで受けた社員の方々の意見はシステムの技術的な面から実用性の面まで多々ありましたが、こういった意見を聞くと、やはり自分達と目の付け所が違っていると感じるが多かったです。実際自分達の中でもあいまいになっていた箇所を鋭く指摘されて、そのときに自分達の認識不足に気づかされたりもしました。まさに現場の空気に触れられた瞬間であったと思います。

このプロジェクトに参加したのは半年という短い期間でしたが、当初の「現場の空気に触れる」という目的は、十分に達成できたと感じました。ただ外側から眺めているだけでなく、一歩中に踏み入れて初めてわかったことが多々ありました。ここで得られた経験を生かして、これからさらにソフトウェア開発に関してもグループでのプロ塾と管理に関しても勉強していきたいと思います。

1年間の間、本当にお世話になりました。これからは是非、よろしくお願いいたします。