

niconico ライブ・プロジェクト 最終報告書

PM

高橋 宏子 (東京海上日動システムズ)

学生

川口 将司 (環境情報学部・4年)

鳴澤 智行 (環境情報学部・4年)

渡辺 まや (環境情報学部・2年)

目次

1	プロジェクトの定義と計画 -----	p.3
1.1	プロジェクト定義書 (p.3~)	
1.2	プロジェクトマネジメント計画書 (p.8~)	
2	プロジェクト実績報告 -----	p.17
3	クライアントへの提案 -----	p.22
3.1	ヒアリングの実施 (p.22~)	
3.2	モックアップ・オブジェクト作成の効果 (p.28~)	
3.3	システム提案書 (p.29~)	
4	システム構成 -----	p.35
5	設計 -----	p.36
5.1	データベースの設計 (p.36~)	
5.2	画面の設計 (p.40~)	
5.3	個別関数の定義 (p.54~)	
6	映像技術について -----	p.59
7	テスト -----	p.70
8	クライアントからの評価 -----	p.90
8.1	クライアントテストとアンケートへの回答 (p.90~)	
8.2	今後の展望 (p.92~)	
9	プロジェクト総括 -----	p.93
10	付録 -----	p.94

1 プロジェクトの定義と計画

1.1 プロジェクト定義書

水槽コミュニケーションシステム プロジェクト定義書

~ version 1.2 ~

作成日：2007/10/15

更新日：2007/10/31

作成者：高橋 宏子

更新履歴

版	更新内容	更新者	更新日
1.0	新規作成	高橋	2007/10/15
1.1	本書の構成を変更	高橋	2007/10/25
1.2	以下について補記 ・ 顧客要求事項 ・ プロジェクト・スコープ ・ 制約事項	高橋	2007/10/31

1 はじめに

本書は、水槽のライブ映像配信と各種コミュニケーションを行うことができる Web アプリ（仮称「水槽コミュニケーションシステム」）作成プロジェクトのプロジェクト定義書である。

2 プロジェクト概要

2.1 プロジェクト名

プロジェクト名は「niconico ライブ」とする。

2.2 顧客

松尾壮紘氏

2.3 ステークホルダー

- ・ 松尾氏
- ・ 熱帯魚オーナー仲間
- ・ 大岩研究室（大岩教授、松澤氏、橋山氏）

3 プロジェクトの目的と妥当性

3.1 顧客環境の課題

松尾氏は、熱帯魚を観賞するのが好きだが、自宅には水槽を置くスペースがなく、かわりに水槽を持っている友人に自分の魚を飼育してもらっている。しかし、友人宅までの距離が遠いこともあり、一度も様子を見にいけていない。

3.2 プロジェクト立上げの経緯

友人宅へ頻繁に出向いて自分の魚の様子を見に行くのが困難である松尾氏の「離れたところから、インターネットを介して熱帯魚の様子が見たい」「ただ見られるだけでは面白くないので、飼い主間のコミュニケーションを促進するような何か欲しい」という欲求を満たすべく、今回本システムを作成するに至った。

4 プロジェクトの目的・妥当性

4.1 プロジェクトの目的

本プロジェクトの目的は、水槽コミュニケーションシステムを予定された期間・コストで作成し、顧客に喜んで（ニコニコして）いただくことである。

4.2 プロジェクトの妥当性

離れた場所の様子を知り、離れた場所にいる人間とコミュニケーションをとるために、Web の技術を利用することは非常に有効である。また、本来人間が側にいない状況でしか見られない魚の習性を、カメラを通して見られる可能性がひろがるので、本システムを作成することには非常に意味がある。

5 プロジェクト要求事項

5.1 顧客要求事項

- (1) 遠隔地にある水槽のライブ映像配信と、熱帯魚オーナー同士の各種コミュニケーションを可能とするシステムの構築
- (2) 熱帯魚オーナーではない一般ユーザーについても、水槽の様子や熱帯魚の情報、オーナーの書く日記を閲覧できると良い

5.2 成果物に対する要求事項

顧客へ提出する物件は以下のとおりとする。

- ・ 水槽コミュニケーションシステム
- ・ 運用マニュアル

ただし、上記納品物件はプロジェクト発足時のものであり、納品時に変更される可能性がある。

5.3 プロジェクト・スコープ

5.3.1 スコープに含める作業

- ・ インターネットを介して遠隔地にある水槽の様子を閲覧できる機能の提供
- ・ 熱帯魚の飼い主同士が各種コミュニケーションを図れる機能の提供

5.3.2 スコープから除外する作業

- ・ 導入する Web カメラ製品の選定
- ・ 本稼働後のヘルプデスク対応

5.4 制約事項

本プロジェクトにおける制約事項は以下のとおりである。

- ・ 顧客が用意するサーバの OS は、Linux である。
- ・ 新たに Web カメラを導入する資金はないので、既に顧客が所持しているものを利用する。
- ・ 今回 Web カメラを設置する場所は、個人宅ではなく、松尾氏の友人が所属する会社になる。セキュリティを考慮すると、音声を拾うことは不可である。

5.5 受け入れ基準

顧客要求を満たすシステムを構築し、ユーザテストに合格すること。

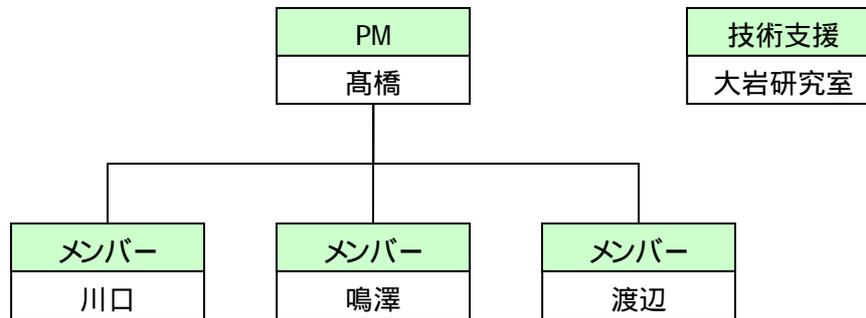
6 期間

プロジェクトの期間は、2007 年 10 月 4 日から 2008 年 1 月 31 日までとする。

7 スケジュール

別紙参照

8 開発体制



PM 高橋宏子（東京海上日動システムズ株式会社）
メンバー 川口将司（環境情報学部4年）
鳴澤智行（環境情報学部4年）
渡辺まや（環境情報学部2年）

9 会議体

全体：毎週木曜日 3限

チームミーティング

目的：PM を交えたチームミーティングを実施する

時間：毎週木曜日 4限

出席者：原則全員

チームミーティング

目的：PM を除いたメンバー間での共同作業を実施する

時間：毎週火曜日 3限

出席者：PM を除くメンバー

メンバー間のコミュニケーションには、メーリングリスト、Wiki（ ）を主に用いる。
（ <http://eagle.crew.sfc.keio.ac.jp/wiki/2007f/wiki.cgi?page=FrontPage> ）

以上

Master Schedule

プロジェクト	niconicoライブ	作成	更新	承認	添付資料	page	0010
		10/15	12/5		<input type="checkbox"/> 有		
		高橋 宏子	高橋 宏子		<input checked="" type="checkbox"/> 無		

	10月				11月				12月				1月			
	8	15	22	29	5	12	19	26	3	10	17	24	7	14	21	28
1	要件確認・分析												年末年始			
2					設計											
3						実装										
4									テスト							
5															評価	
6																移行

備考

2007/10/25 :
月の「1週目」「2週目」といった表記を、具体的な日付(その週の月曜日)表記に変更。
また、スケジュール内に「年末年始」を明記。

2007/11/07 :
1度目の反復と、2度目の反復について、期間を調整。 WBS一覧から判断した

2007/12/05 :
1度目の反復について、実装期間を一週間延長。 単体テストも「実装」フェーズに含めるので

1.2 プロジェクトマネジメント計画書

水槽コミュニケーションシステム プロジェクトマネジメント計画書

~ version 1.2 ~

作成日：2007/10/31
作成者：高橋 宏子

更新履歴

版	更新内容	更新者	更新日
1.0	新規作成	高橋	2007/10/31
1.1	以下について補記 ・ システム化対象範囲および対象外範囲 ・ コスト計画 ・ 学習計画	高橋	2007/11/07
1.2	誤字を修正	高橋	2007/11/15

1 本書の目的

本ドキュメントは、「niconico ライブ」プロジェクトにおけるプロジェクト・マネジメント計画を記述したものである。

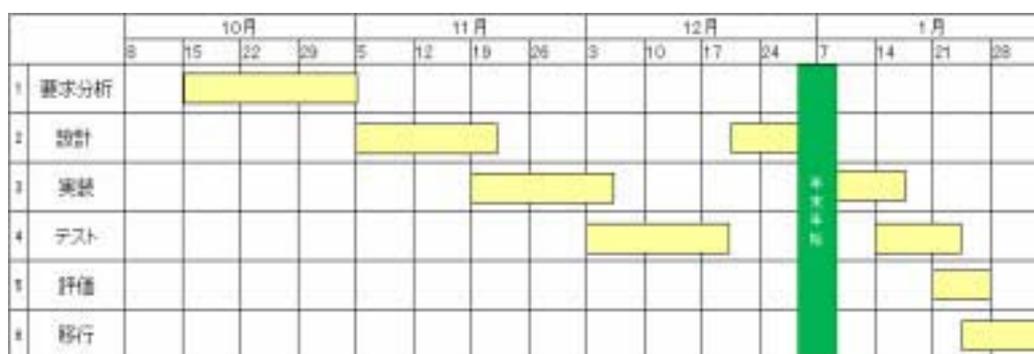
2 プロジェクト概要

本プロジェクトにおいては、水槽のライブ映像配信と各種コミュニケーションを行うことができる Web システム構築を目指す。

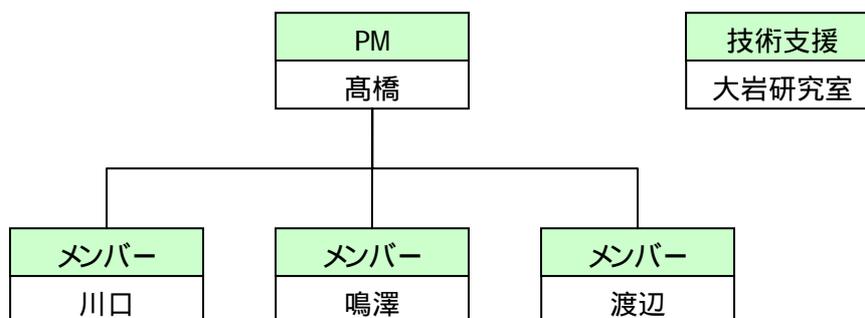
3 プロジェクト構成

3.1 プロジェクトの期間

2007年10月4日から2008年1月31日までとする。



3.2 開発体制

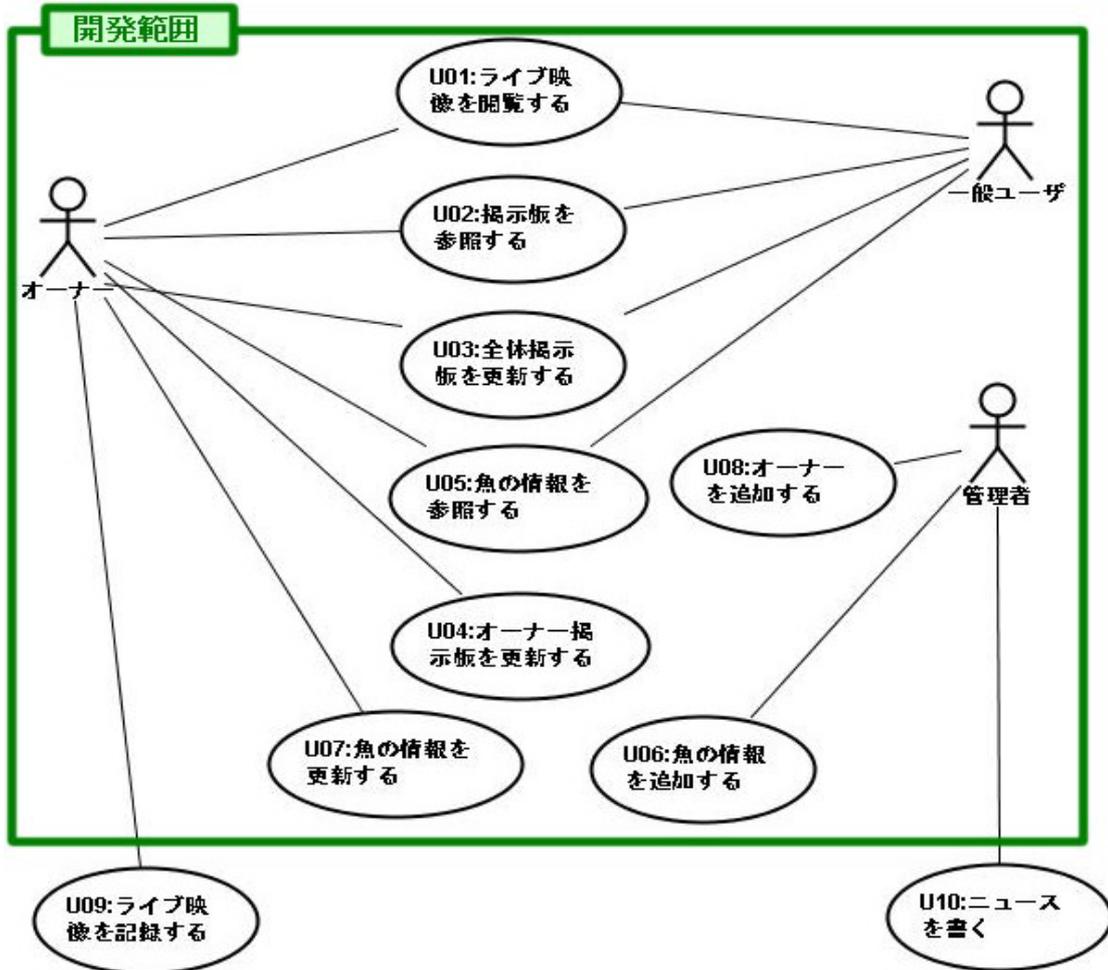


PM 高橋宏子（東京海上日動システムズ株式会社）
メンバー 川口将司（環境情報学部4年）
 鳴澤智行（環境情報学部4年）
 渡辺まや（環境情報学部2年）

4 システム化対象範囲および対象外範囲

本プロジェクトにおけるシステム開発範囲を、下図に示す。

詳細は、クライアントに提出する「システム提案書」を参照のこと。



5 品質計画

5.1 成果物品質

ドキュメント名	作成	承認	備考
プロジェクト定義書	高橋	大岩研	提出済（変更がある際は随時更新）
WBS（スケジュール）	高橋	大岩研	提出済（変更がある際は随時更新）
システム提案書	メンバー	顧客	
プロジェクト計画書	高橋	大岩研	本書（プロジェクトマネジメント計画書）
画面仕様書	メンバー	顧客	画面毎の機能が確認できること
テスト計画書	メンバー	大岩研	モジュール単位でテストケースを作成
ソースコード	メンバー	大岩研	内容コメントが記述されていること
最終報告書	メンバー	大岩研	

5.2 テスト計画

テスト名	担当者	目的	品質
単体テスト	メンバー	画面毎の機能確認	テスト計画書、各自の担当部分以外を持ち回る
総合テスト	メンバー	システム全体での挙動確認	画面仕様書
稼働確認	メンバー 顧客 ユーザ	本番環境での挙動確認	クライアントである松尾様と共に挙動確認を実施する

* 一般ユーザによるテストは実施しない。

5.3 評価

- (1) 顧客およびユーザに対し、アンケートを実施する。アンケート結果が、事前に顧客と協議の上で決定した総合ポイントに達していることを、受け入れの最低条件とする。
- (2) 顧客に対し、インタビューを実施する。ビデオを撮影し、顧客の反応を以って受け入れに前向きであるかどうか判断する。

6 スケジュール計画

開発モデルは「反復型」を採用する。

別紙（WBS 一覧）参照

7 システム管理計画

納品後の保守、管理については顧客側で実施し、本プロジェクトメンバーはそれに関わらないものとする。

8 リスク管理

想定し得るリスクと対策について、以下に示す。

リスク 1	
リスク内容	PM が遠隔地での作業であり、週 1 日の参加のため、顧客やメンバーとのコミュニケーションが不足する可能性がある。
対策	PM を除いたメンバーによるミーティングを週に一度実施し、メンバー間のコミュニケーションを図る。 作業時間を考慮した適切なスコープの定義を行う。 コミュニケーションツールを使用し、迅速に対応する。

リスク 2	
リスク内容	他の授業により、メンバーに割り当てられたプロジェクト作業が計画通りに進まない可能性がある。
対策	プロジェクト計画を通じてメンバーの作業時間を事前に伝え、各メンバーの作業計画を立てやすくする。 作業時間を考慮した適切なスコープの定義を行う。

リスク 3	
リスク内容	顧客が別プロジェクトの PM を兼ねているので、本プロジェクトにあまり多く携われない可能性がある。
対策	問題があれば、PM ミーティングの場で相談する。 場合によっては別の人間を窓口としていただくことを検討する。

9 コスト計画

本プロジェクトの人的リソースについては、作業時間をもってコストとする。

作業時間の見積りを以下に示す。(単位：時間)

	2007/10	2007/11	2007/12	2008/01	2008/02	合計
川口将司	35.0	40.0	45.0	45.0	0.0	165.0
鳴澤智行	35.0	40.0	45.0	45.0	0.0	165.0
渡辺まや	35.0	40.0	45.0	45.0	0.0	165.0
高橋宏子	30.0	30.0	20.0	15.0	0.0	95.0
合計	135.0	150.0	155.0	150.0	0.0	590.0

* スケジュール上の各フェーズ、およびメンバーの作業分担により算出した。

* 各メンバーの授業参加、PM の管理に要するタスク時間は除いている。

10 学習計画

10.1 チーム目標

期日内にシステムを構築し、チーム開発としての技術、ノウハウを習得する。

10.2 個人目標

氏名	目標
川口将司	Web カメラという使い慣れないデバイスを利用した技術を身につける。 Linux サーバー用途のスキルを上げる。 開発プロセスを気にし続けて、マネジメントスキルに活かす。
鳴澤智行	チームでのプログラミングを経験する。 データベース系の扱いを学ぶ。 Web アプリケーションを作る経験をする。
渡辺まや	ひとつのプロジェクトに取り組むうえでの一連の流れをつかむ。 自らのプログラミング能力・技術の向上をはかる。 PM になって、マネジメントについてもできる限り学んでいく。
高橋宏子	プロマネの経験を積む。 Wiki 等のコミュニケーションツールを利用したプロジェクト管理の有用性を検証する。 遠距離でのチーム開発を機能させる。

以上

成果物	(niconicoライブ)WBS一覧	作成	更新	承認
		10/25	1/31	
		高橋 宏子	高橋 宏子	

大項目	中項目	項番	項目	予定			実績			先行	担当	10月					11月				12月				1月					
				開始日	日数	終了日	開始	終了	0%			10/1	10/8	10/15	10/22	10/29	11/5	11/12	11/19	11/26	12/3	12/10	12/17	12/24	12/31	1/7	1/14	1/21	1/28	
要件確認 分析	要求分析	1	基本計画	10/04	30日	11/02	10/04					工程	基本計画(10/4 - 11/2)																	
		1-1	顧客ヒアリング	10/04	15日	10/18	10/04	10/18		100%		メンバー	10/18																	
		1-2	熱帯魚調査	10/18	8日	10/25	10/19	10/24		100%		渡辺	10/18 10/25																	
		1-3	Webカメラ性能調査	10/18	8日	10/25	10/19	10/25		100%		鳴澤	10/18 10/25																	
		1-4	Webカメラ技術アセスメント	10/25	7日	10/31	11/01	11/15		100%		川口	10/25 10/31																	
	プロジェクト計画	1-5	提案用モックアップ作成	10/11	8日	10/18	10/11	10/17		100%		渡辺	10/11 10/18																	
		1-6	工数見積もり	10/25	8日	11/01	11/01	11/05		100%		高橋	10/25 11/1																	
		1-7	プロジェクト計画書作成	10/25	9日	11/02	10/25	11/07		100%		高橋	10/25 11/2																	
	システム提案	1-8	プロジェクト計画書DR	11/02	1日	11/02	11/01	11/08		100%		高橋	11/2 プロジェクト計画書DR																	
		1-9	システム提案書作成	10/18	15日	11/01	11/01	11/07		100%		高橋	10/18 11/1 システム提案書作成																	
1-10		システム提案書DR	11/02	1日	11/02	11/08	11/15		100%		高橋	11/2 システム提案書DR																		
設計	設計	2	設計	11/05	18日	11/22	11/05	11/22				工程	設計 (11/5 - 11/22)																	
		2-1	DB設計	11/05	18日	11/22	11/08	11/15		100%		鳴澤	11/5 11/22																	
		2-2	画面設計	11/12	8日	11/19	11/08	11/21		100%		渡辺	11/12 11/19																	
		2-3	画面項目定義の作成	11/12	8日	11/19	11/08	11/21		100%		渡辺	11/12 11/19																	
	設計	2-4	松尾さんへの確認	11/22	1日	11/22	11/22	11/22		100%		メンバー	11/22 松尾さんへの確認																	
		2@	設計	12/20	10日	12/29	12/24	1/04				工程	設計 (12/20 - 1/4)																	
		2-5	リリース#1のフィードバック	12/20	3日	12/22	12/24	12/24		100%		メンバー	12/20 12/22																	
		2-6	画面仕様書の作成	12/22	7日	12/28	1/04	1/29		100%		メンバー	12/22 12/28																	
実装	実装	3	実装	11/19	18日	12/06	11/19	12/16		2	工程	実装 (11/19 - 12/6)																		
		3-1	Webサーバ構築	11/19	7日	11/25	11/19	11/19		100%		松尾様	11/19 11/25																	
		3-2	データベース構築	11/19	7日	11/25	11/19	11/28		100%		鳴澤	11/19 11/25																	
		3-3	コーディング	11/22	14日	12/05	11/22	12/13		100%		メンバー	11/22 12/5																	
		3-4	単体テスト実施	11/22	14日	12/05	11/29	12/13		100%	3-3	メンバー	11/22 12/5																	
	実装	3-5	ソースレビュー	11/30	7日	12/06	12/13	12/16		100%	3-4	メンバー	11/30 12/6 ソースレビュー																	
		3@	実装	1/07	11日	1/17	1/04	1/21		2@	工程	実装 (1/7 - 1/21)																		
		3-6	コーディング	1/07	8日	1/14	1/04	1/20		100%		メンバー	1/7 1/14																	
[3.6人月]	3-7	単体テスト実施	1/07	8日	1/14	1/04	1/20		100%	3-6	メンバー	1/7 1/14																		
	3-8	ソースレビュー	1/12	6日	1/17	1/21	1/21		100%	3-8	メンバー	1/12 1/17 ソースレビュー																		
	4	テスト	12/03	18日	12/20	11/21	12/24		3	工程	テスト (12/3 - 12/20)																			
テスト	テスト	4-1	テスト環境構築	12/03	7日	12/09	11/21	11/21		100%		松尾様	12/3 12/9																	
		4-2	テスト実施	12/10	8日	12/17	12/16	12/21		100%	4-1	メンバー	12/10 12/17																	
	稼働確認	4-3	稼働確認実施	12/18	3日	12/20	12/20	12/24		100%	4-2	松尾様	12/18 12/20																	
		4@	テスト	1/14	11日	1/24	1/21	1/24		3@	工程	テスト (1/14 - 1/24)																		
		4-4	テスト実施	1/14	8日	1/21	1/21	1/21		100%		メンバー	1/14 1/21																	
		4-5	システムテスト結果報告	1/21	1日	1/21	1/21	1/21		100%	4-4	メンバー	1/21 システムテスト結果報告																	
稼働確認	4-6	稼働確認実施	1/22	3日	1/24	1/21	1/24		100%	4-5	松尾様	1/22 1/24																		
	5	評価	1/21	7日	1/27	1/24	1/31		4@	工程	評価 (1/21 - 1/31)																			
評価	評価	5-1	インタビュー実施	1/21	3日	1/23		11日	0%		メンバー	1/21 1/23																		
		5-2	アンケート実施	1/21	3日	1/23	1/24	1/28		100%		メンバー	1/21 1/23																	
		5-3	評価のまとめ	1/23	5日	1/27	1/28	1/31		100%	5-2	メンバー	1/23 1/27 評価のまとめ																	
移行	移行	6	移行 納品	1/24	8日	1/31	1/26	1/31		5	工程	移行 (1/24 - 1/31)																		
		6-1	インストールマニュアル作成	1/24	4日	1/27	1/29	1/31		100%		メンバー	1/24 1/27																	
		6-2	運用マニュアル作成	1/24	4日	1/27		8日	0%		メンバー	1/24 1/27																		
		6-3	最終報告書作成	1/24	7日	1/30	1/26	1/31		100%	6-2	メンバー	1/24 1/30																	
	6-4	納品	1/31	1日	1/31	1/31	1/31		100%		メンバー	1/31 納品																		

2 プロジェクト実績報告

niconico ライブ プロジェクト完了報告書

作成日：2008/01/31

作成者：高橋 宏子

1 プロジェクト概要

本プロジェクトにおいては、水槽のライブ映像配信と各種コミュニケーションを行うことができる Web システムを構築することとした。

2 プロジェクト構成

2.1 プロジェクト対象期間

2007 年 10 月 4 日 ~ 2008 年 1 月 31 日

2.2 プロジェクトメンバー

PM 高橋宏子（東京海上日動システムズ株式会社）

メンバー 川口将司（環境情報学部 4 年）

鳴澤智行（環境情報学部 4 年）

渡辺まや（環境情報学部 2 年）

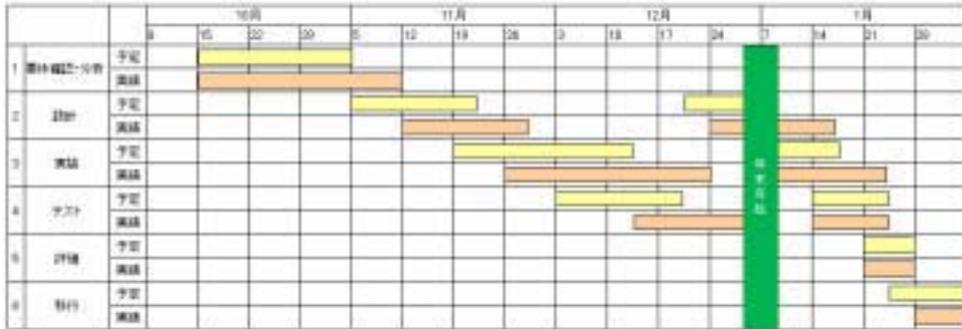
3 完了成果物

ドキュメント名	備考
プロジェクト定義書	本プロジェクトを遂行する背景、目的、概要、スケジュール、体制図を記載
WBS（スケジュール）	本プロジェクトのスケジュール管理を記載
システム提案書	本プロジェクトを遂行することをクライアントへ提示することを目的とする
プロジェクト計画書（プロジェクトマネジメント計画書）	本プロジェクトにおけるシステム化対象、対象外範囲、品質計画、スケジュール計画、システム管理計画、リスク管理計画、見積計画、学習計画、コミュニケーション計画を記載
画面仕様書	画面毎の機能が確認できること
テスト計画書	テストシナリオ/ケースを作成 総ケース数：147、総パターン数：207
ソースコード	総行数：1730、総ステップ数：1164 [参考]テストコードの行数：251、スタブの行数：284
インストールガイド	技術者向け
プロジェクト完了報告書	本ドキュメント（スケジュール計画、コスト計画に対する実績を記載）

4 スケジュール

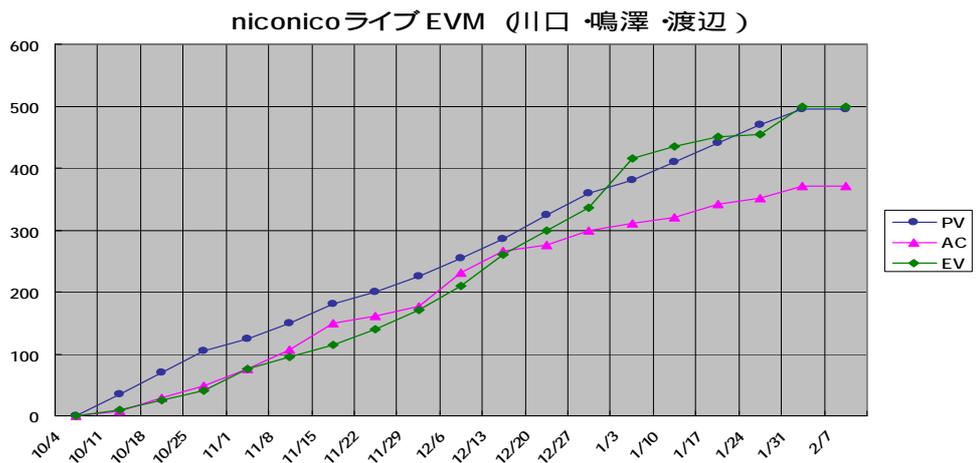
4.1 計画と実績

マスタスケジュールレベルでのスケジュール実績を以下に示す。

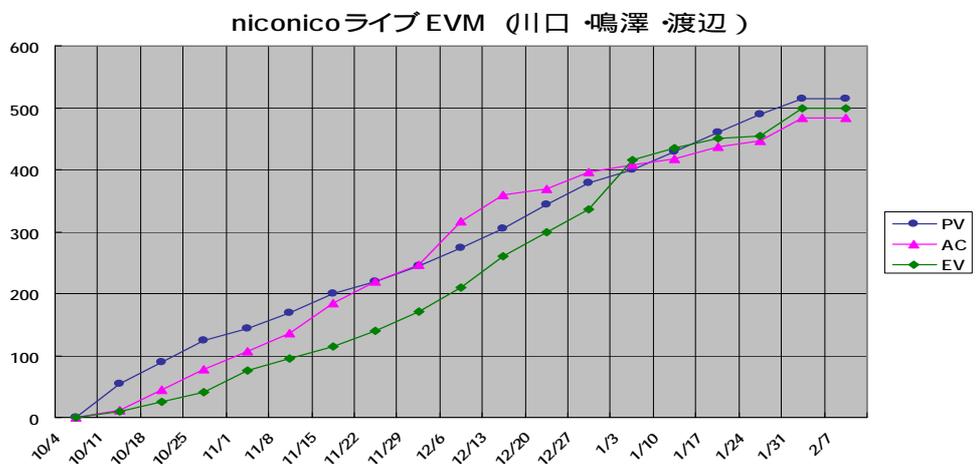


EVM については、直接コストのみのパターン、間接コストを含むパターンの 2 通りについて算出した。

直接コストのみ



間接コストを含む



4.2 考察

1 回目の反復においては、クライアントの要求を引き出してシステムに落とし込むまでに非常に多くの期間（時間）を要し、設計も念入りに行ったことで、計画よりもスケジュールが遅延してしまった。しかし、1 回目にじっくり取り組んだことで、2 回目の反復におけるキャッチアップが可能となり、最終的にはほぼスケジュール通りにプロジェクトを遂行できた。

2 つの EVM グラフを見比べてみると、本プロジェクトにおいては間接コストが多かったことがわかる。どこまでを直接・どこからが間接、という明確な定義がなかったので、例えば「勉強」という時間をどうコスト計上すべきか非常に悩んだ。今回は、本プロジェクトに携わる上で必要な勉強については直接、それ以外の勉強（例えば「テストとは？」のようなこと）については直接コストとして考えた。トータル的には計画通りのコストでプロジェクトを遂行できたと言えるであろう。

5 コスト計画および実績

（単位：時間）

項目	計画	実績	差分
要件確認・分析	160	126.5	33.5
設計	105	63.5	41.5
映像配信：技術調査、実装	90	136	+ 46
PHP アプリ：実装、単体テスト	140	113	27
総合テスト	73	45	28
評価・移行（稼働確認テストを含む）	22	16	6
各種資料作成		55.5	+ 55.5

6 リスク計画対応実績

リスク 1	
リスク内容	PM が遠隔地での作業であり、週 1 日の参加のため、顧客やメンバーとのコミュニケーションが不足する可能性がある。
対応実績	必要に応じて、PM を除いたメンバーによるミーティングを実施し、メンバー間のコミュニケーションを図った。また、Wiki 等のコミュニケーションツールを使用し、迅速に対応するよう努めた。

リスク 2	
リスク内容	他の授業により、メンバーに割り当てられたプロジェクト作業が計画通りに進まない可能性がある。
対応実績	作業内容と作業にかかる時間を事前に伝え、各メンバーの作業計画を立てやすくするように努めた。しかし、見積不足やコミュニケーション不足により浸透しなかった。 作業時間を考慮した適切なスコープの定義を行ったので、大きく計画から狂うことはなかった。

リスク 3	
リスク内容	クライアントが別プロジェクトの PM を兼ねているので、本プロジェクトにあまり多く携われない可能性がある。
対応実績	問題があれば PM ミーティングの場で相談したり、場合によっては別の人間を窓口としていただく等の検討をする予定でいたが、そのような心配はまったくなかった。

7 学習計画対応実績

7.1 チーム目標

【計画】

期日内にシステムを構築し、チーム開発としての技術、ノウハウを習得する。

【実績】

メンバーの努力およびクライアントの多大なる力添えのおかげで、期日内にシステムを納めることができた。明確な担当わけを行いながらも、チームとしての連携を心がけ、チーム開発の楽しさや難しさを実感することができた。

7.2 個人目標

個人毎の学習成果報告書を参照

以上

3 クライアントへの提案

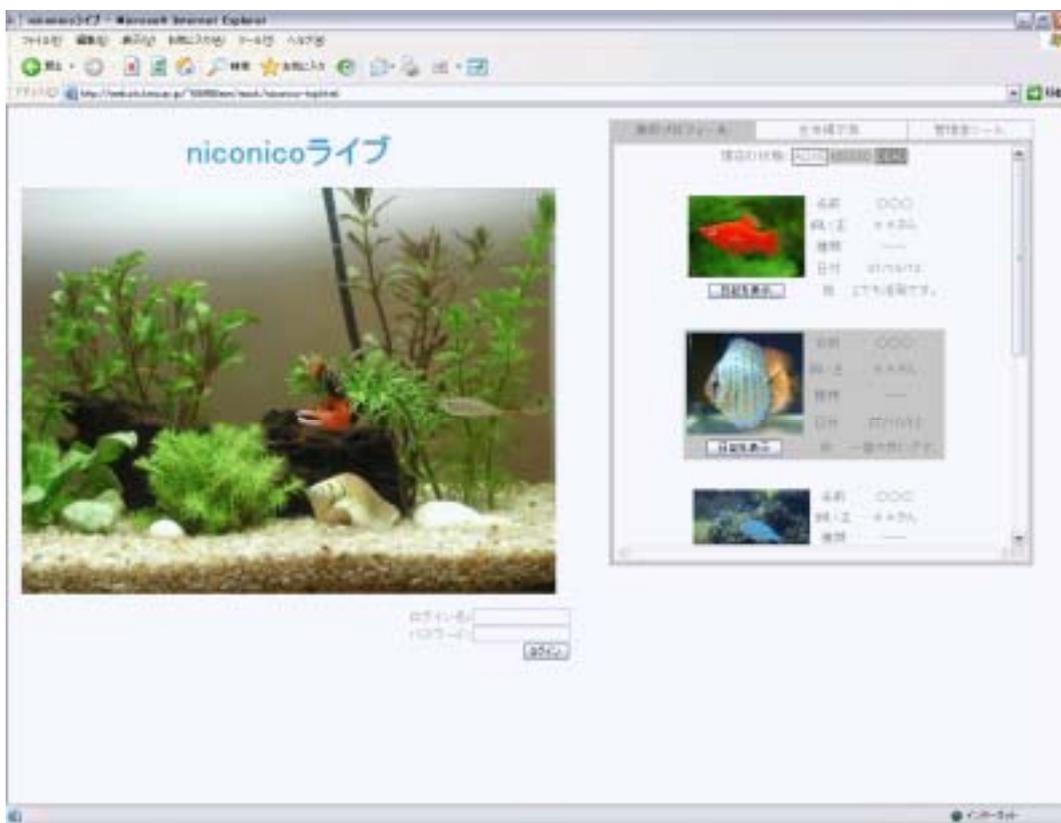
3.1 ヒアリングの実施

クライアントである松尾さんへは、計3回のヒアリングを実施した。まずは、最初のヒアリングで、以下について確認した。

- ・ システムを使って実現したいこと
- ・ 動作環境について
- ・ 「必要な機能」と「あったらいい機能」について
- ・ 第1回クライアントテストについて

これらのヒアリング結果をもとに、メンバー内で「どのようなシステムを作るか」について論議をした。しかし、どうしても「松尾さんに言われたから作る」という思考になってしまい、「松尾さんはなぜこのシステムを必要としているのか」「これが出来ると何が面白いのか」という点についての理解が欠けていた。このまま「言われたとおりのモノを作る」だけでは自分たちの色を出すことができない、と気がつき、前述のような点について考える上でのイメージを湧かせるために、モックアップ・オブジェクト (html ベースのプロトタイプ) を作成することにした。

図表 3.2-1 モックアップ・オブジェクトの一例



2回目のヒアリングにおいては、作成したモックアップ・オブジェクトを用いて、今回作成するシステムが「なぜ必要なのか」「何が面白いのか」のヒントを得るべく松尾さんと議論を重ねた。その中で、以下のような意見をいただくことができた。

- ・ 一箇所にしかない水槽を複数人間が共有し、眺めることが出来るのは魅力。
犬猫とは異なり、熱帯魚は「眺められる」ことに価値を持つペットである。
- ・ 人がそばに居ると（カメラを通してしか）見られない魚の習性がある。
人の目があると、魚は自然に振舞ってくれないことが多々ある。

話の中で、松尾さんからは「プレコストムス」という熱帯魚を紹介していただいた。この魚は、水槽に付いた藻を食べる魚であるが、人目を嫌い、決して人前には出てこないという習性を持つ。このように、人の目があると見ることができない、あるいは稀にしか見ることができない面白い魚の習性が他にもないだろうか、もしあるとすれば、そのような習性がカメラを通じて目に出来るかもしれないということこそ、システム作成の価値につながるのではないかと考えた。そこで、熱帯魚についての調査を実施した。

【成果物：熱帯魚調査報告書】

熱帯魚調査報告書

niconico ライブプロジェクト

1. 調査対象

- (1) 熱帯魚ショップ：スプラッシュ宮前平店
- (2) 熱帯魚飼育者(現在飼育中・過去飼育経験ありの両者)：数名

2. 調査担当

プロジェクトメンバー 渡辺まや

3. 調査目的

- (1) 本プロジェクトで作成しようとしている「水槽コミュニケーションシステム(仮称)」の要件分析におけるヒントを得るため、プロジェクト対象となる熱帯魚・熱帯魚の飼育についての理解を深める。
- (2) 本システム作成により生まれる利点の重要性について調べる。
- (3) 実際に熱帯魚ショップへ赴くことで、熱帯魚観賞の際に付随する事柄を調べる。

4. 調査方法

- (1) 熱帯魚ショップ
スプラッシュ宮前平店に赴き、スタッフの方に直接インタビューを行う。
- (2) 熱帯魚飼育者
 - a. プロジェクトメンバーの友人の飼育者に文書の形で質問を行う。
 - b. ソーシャルネットワーキングサイト mixi の熱帯魚コミュニティにて質問項目を書いたトピックを立て、返信を集計する。

5. 調査結果

2・3 ページ参照。

(1) 熱帯魚ショップ

質問目的	質問内容		回答
飼育における負担を知る	維持費はどれくらいかかるか		60cm 水槽(容積 60L) : 2000 ~ 3000 円 / 月 1.5m 水槽(容積 600L) : 30000 円 / 月
	水槽の大きさはどれくらいか		当店で扱っているものは 20cm ~ 3m まで
	世話をすることは大変か		<ul style="list-style-type: none"> ・現在では自動餌やりや温度管理の自動システムが存在するので、世話は特に必要がない。 ・病院や会社などでは、月に 1 度業者がメンテナンスをしにいくという方法をとっているところが多い。
熱帯魚について知る	熱帯魚のサイズの幅はどれくらいか		<ul style="list-style-type: none"> ・成長しても 2cm しかないものから、記録としては 3m になるものまでいる。 ・お客さんが飼育しているものでの最大は 1m20cm 程度。
	熱帯魚の特徴	泳ぎ方	<ul style="list-style-type: none"> ・ハタタテハゼなど数種類の熱帯魚は斜め上をむいて泳ぐ。 ・つがい・群れで・単独で泳ぐものがある。 ・上部・中部・下部でしか泳がない魚がいるため、バランスよく飼うことで水槽の中に均等に魚が泳ぐよう配置するとよい。
		行動	<ul style="list-style-type: none"> ・キッシンググラミー：ケンカをするときにキスをする
		混泳	<ul style="list-style-type: none"> ・肉食の魚もいるので注意が必要。 ・フグなどは好奇心が旺盛なため、他の魚のひれをかじることも。
		人慣れ	<ul style="list-style-type: none"> ・天然魚は人に慣れていないが、養殖魚は人慣れしているので人が近くにいてもあまり隠れることはない。
熱帯魚飼育の楽しみを知る	飼育していて一番の楽しみは何か		<ul style="list-style-type: none"> ・楽しいというよりは生活の一部であると考えている。 ・餌をやるときに魚が一番動くので楽しい。 ・ブリーディングなども楽しい。 ・水槽の水をかえたとき、魚が綺麗に見えるので嬉しい。

(2) 熱帯魚飼育者

質問目的	質問内容	回答
飼育における負担を知る	自分の魚の特技	<ul style="list-style-type: none"> ・費用がかかる。(飼育にはつきものであるから仕方ないとは考えている) ・やたらと繁殖してしまう。
熱帯魚について知る	自分の魚の特技	<ul style="list-style-type: none"> ・ジャンプをする。(魚の種類：アロワナ) ・水槽をたたくと餌の合図と知って水面へ寄って来る。(魚の種類：グッピー)
	飼っている間に起きた出来事	<ul style="list-style-type: none"> ・フィルターに吸われた稚魚が、濾過槽で成長していた。 ・熱帯魚内でいじめのようなものがあった。 ・温度計が壊れて旅行中に熱帯魚が煮えてしまった。
熱帯魚飼育の楽しみを知る	飼育していて一番の楽しみは何か	<ul style="list-style-type: none"> ・熱帯魚の成長や繁殖。 ・種類によって違う魚の習性を目の前で見られたときに感動できること。 ・魚たちの子育て。(ペアで育児をするディスカスやシクリッド) ・ブリーディング。(グッピーなど) ・元気良く泳いでいる魚をみて元気がもらえること。
本システムの意義への調査	自分が外出している際の魚の様子が気になるか	<ul style="list-style-type: none"> ・新しい魚を入れたときにはどうなっているか気になる。 ・魚の調子が悪ければ気になる。 ・地震で水槽が揺れたとき、水槽から飛び出していないか・水が減っていないか気になる。 ・特に気にしない。

以上のアンケート結果から

- ・本システム作成により水槽の設置スペース・維持費が大幅に削減されると見込まれる。
- ・外出時に魚の様子が気になるという人の多さから、本システム作成は有意義である。

熱帯魚ショップに赴いてみてわかったこと

- ・水槽の音はたいして気にはならないが、やはりその音により魚がはねる音などはほとんど聞こえることがなかった。
映像に加えて音を配信する必要がない。
- ・水槽内の物体(水草や流木・岩など)が多いと魚は見難く、自然と視点を上下左右に動かしてしまう。
視点切り替えの必要性有り。
しかし、魚の泳ぎ方(上部中部下部を泳ぐ)の特徴を利用すれば、カメラの視点も固定に調節可能であるかもしれない。

以上

メンバー同、熱帯魚についての理解を深めた上で、今回作成するシステムの開発スコープ定義を実施するにあたっての調査をするため、3度目のヒアリング（最終ヒアリング）を実施した。ヒアリングの結果を受け、システム化対象および対象外の機能を洗い出し、「システム提案書」を作成、松尾さんに承認をいただく運びとなった。

3.2 モックアップ・オブジェクト作成の効果

今回、モックアップ・オブジェクトを作ったことには、クライアントのより細かな（UI設計レベルでの）要求が聞けたこと、クライアントとメンバー間で「もっとこうした方がいいのではないか」という具体的な話ができたと、モックアップ・オブジェクトを作成する中でメンバーのシステム作成意欲が上がったことなど、大変大きな効果があったと考える。

後に、松尾さんから「モックアップを作ってくれたのは非常に良かった」とのご意見をいただいている。

3.3 システム提案書

承認日	承認者
2007/11/15.	

水槽コミュニケーションシステム システム提案書

1 本書の目的

本ドキュメントは、「niconico ライブ」プロジェクトの成果物である、水槽コミュニケーションシステム（仮称：水槽のライブ映像配信と各種コミュニケーションを行うことができる Web システム）の提案を記述したものである。

2 背景

松尾様は、熱帯魚を觀賞するのが好きだが、自宅には水槽を置くスペースがなく、かわりに水槽を持っている友人に自分の魚を飼育してもらっている。しかし、友人のところまでの距離が遠いこともあり、一度も様子を見にいけていない。

友人のところへ頻繁に出向いて自分の魚の様子を見に行くのが困難である松尾様の「離れたところから、インターネットを介して熱帯魚の様子が見たい」「ただ見られるだけでは面白くないので、飼い主間のコミュニケーションを促進するような何かが欲しい」という欲求を満たすべく、今回本システムを作成するに至った。

3 要求の分析

松尾様とご友人（本システムの「ユーザ」にあたる方）の要求を分析するために、プロジェクトメンバーが熱帯魚についての調査を実施した。これについては、別紙「熱帯魚報告書」をご参照いただきたい。

4 用語の定義

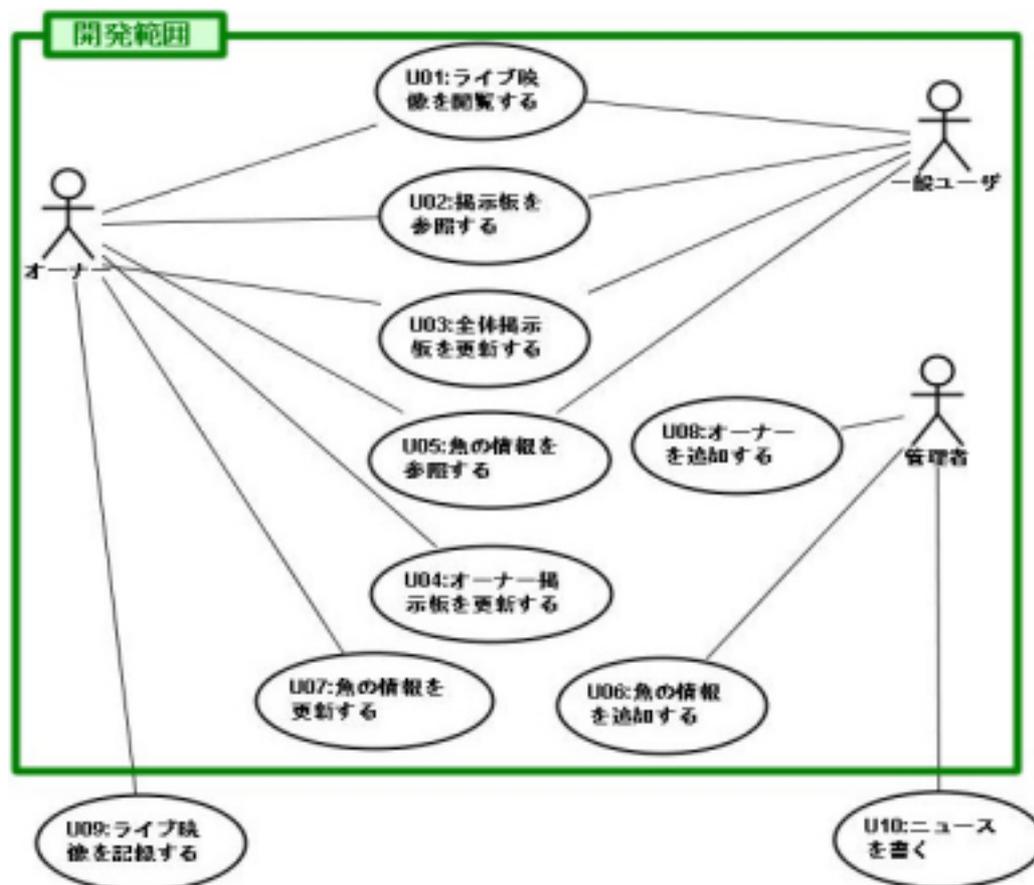
本システムにおける用語の定義は、下表のとおりである。

用語	説明
オーナー	熱帯魚の飼い主。水槽管理者の持つ水槽で、自分の熱帯魚を飼育してもらっている。
管理者	オーナーの熱帯魚を格納する水槽の持ち主であり、本システムの管理者を兼ねる。
一般ユーザ	オーナーでも水槽管理者でもない、一般のユーザ。

5 システムの機能

5.1 開発スコープ

開発期間とヒアリング時にお伺いした優先順位を考慮し、開発範囲を以下の図のとおりとする。



5.2 概要

「管理者」、「オーナー」、「一般ユーザ」について、それぞれができることを以下の表にまとめる。

	管理者	オーナー	一般ユーザ
U01: ライブ映像を閲覧する			
U02: 掲示板を参照する			
U03: 全体掲示板を更新する			
U04: オーナー掲示板を更新する			
U05: 魚の情報を参照する			
U06: 魚の情報を追加する			
U07: 魚の情報を更新する			
U08: オーナーを追加する			

6 画面イメージ

以下のイメージは確定ではない。画面デザインは、画面仕様書を作成する際に決定する。



7 レビューおよびテスト実施方法

7.1 レビュー

本プロジェクトにおいては下表のとおり成果物を作成し、それぞれをレビュー・承認いただくこととする。

成果物名	作成	承認	備考
プロジェクト定義書	PM	大岩研	提出済（変更がある際は随時更新）
WBS（スケジュール）	PM	大岩研	提出済（変更がある際は随時更新）
システム提案書	メンバー	松尾様	本書
プロジェクト計画書	PM	大岩研	
画面仕様書	メンバー	松尾様	画面毎の機能が確認できること
テスト計画書	メンバー	大岩研	
ソースコード	メンバー	大岩研	内容コメントが記述されていること
最終報告書	メンバー	大岩研	

7.2 テスト

本プロジェクトにおいては下表のとおりテストを実施することとする。

テスト名	担当者	目的	備考
単体テスト	メンバー	画面毎の機能確認	テスト計画書、各自の担当部分以外を持ち回る
総合テスト	メンバー	システム全体での挙動確認	画面仕様書を用いる
稼動確認	メンバー 松尾様 ユーザ	本番環境での挙動確認	クライアントである松尾様と共に挙動確認を実施する

* 一般ユーザによるテストは実施しない。

8 松尾様にさせていただく作業

本プロジェクトにおいて、松尾様に実施いただく作業と時期については、下表のとおりである。

作業	概要	時期
要求分析	システム提案書の承認	11/15 まで
サーバ環境設定	DBMS のインストール、各種設定を行う。 詳細の実施内容については、別途ご連絡	11/20 まで
ユーザテスト	クライアント・ユーザ視点での稼動確認テスト	12/18 ~ 12/20 1/22 ~ 1/24
テスト結果通知	テストに対するフィードバック	12/20 ~ 12/22 1/24 ~ 1/26
検収	検収および検収結果報告	1/31 まで

9 受け入れ条件

本システムを松尾様に受け入れていただく際には、以下の条件を満たしているものとする。

- ・ 松尾様およびユーザの皆様が、水槽のライブ映像を閲覧できていること
- ・ 熱帯魚オーナー間のコミュニケーションができていること

また、受け入れるかどうかについては、システム稼動確認テスト後、松尾様に判断していただく。

10 納品物

納品する物件は以下のとおりとする。

- ・ 水槽コミュニケーションシステム・ソースコード
- ・ 運用マニュアル
- ・ インストールマニュアル

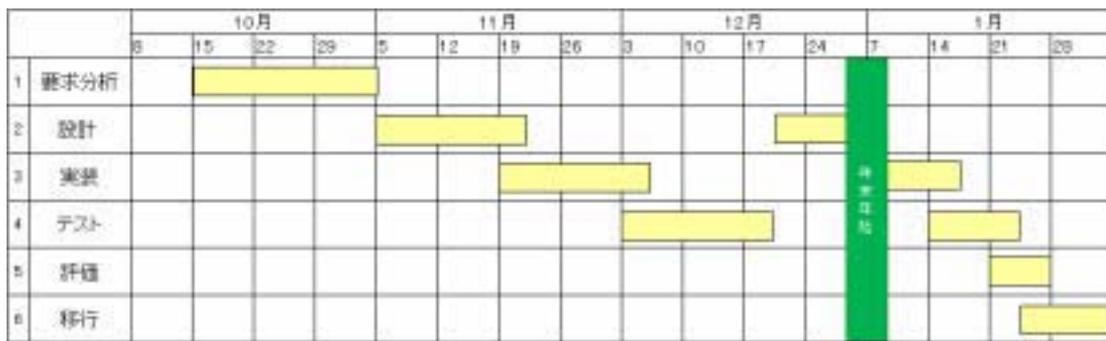
11 制約事項

本プロジェクトにおける制約事項は以下のとおりである。

- ・ クライアントが用意するサーバの OS は、Linux である。
- ・ 新たに Web カメラを導入する資金はないので、既にクライアントが所持しているものを利用する。

12 スケジュール

本システムの開発期間は、2007 年 10 月 4 日から 2008 年 1 月 31 日までとする。



以上

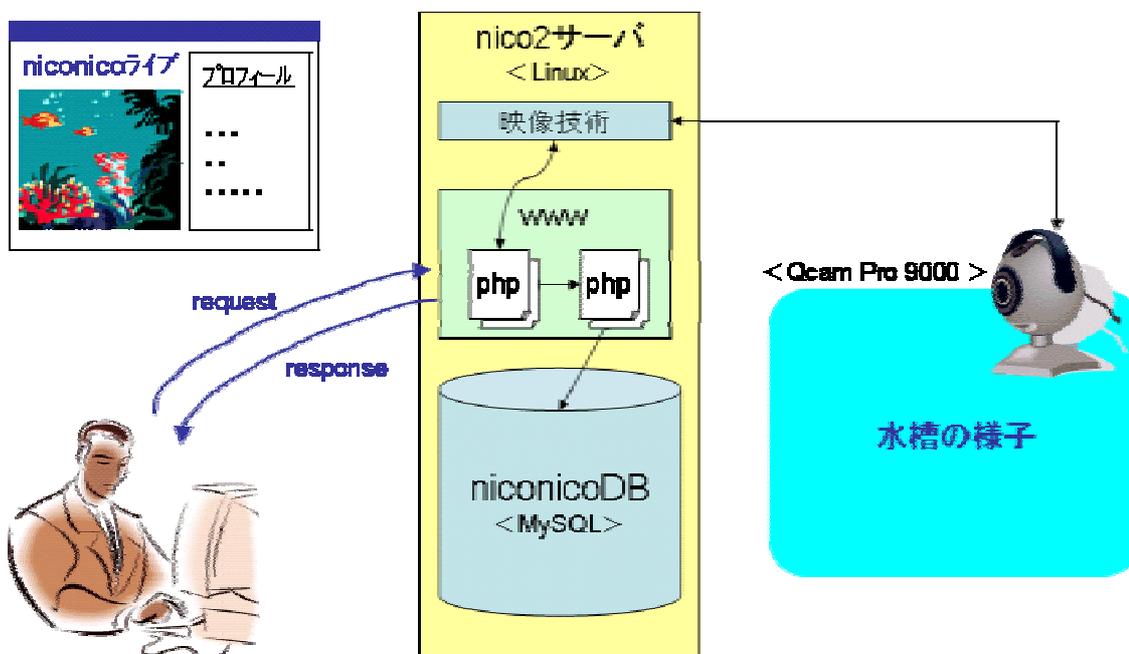
4 システム構成

水槽コミュニケーションシステム「アクアライブ」のシステム構成は、以下とおりである。

図表 4-1 nico2 サーバ仕様

OS	Cent OS 5.1
Web サーバ	Apache 2.0
DBMS	MySQL 5.0.22
開発用言語	Php 5.1.6
Web カメラ	Logicool「Qcam Pro 9000」

図表 4-2 システム構成図



この中の「映像技術」の部分については、プロジェクトの初期から中盤以降まで、長期間にわたりリスクとしてついてまわってきた。結果として、クライアントから要望の強かった「オール Linux での映像配信」を実現することができず、間に Windows サーバを配置することになった。

これについては、「6 映像技術について」で詳しく触れることにする。

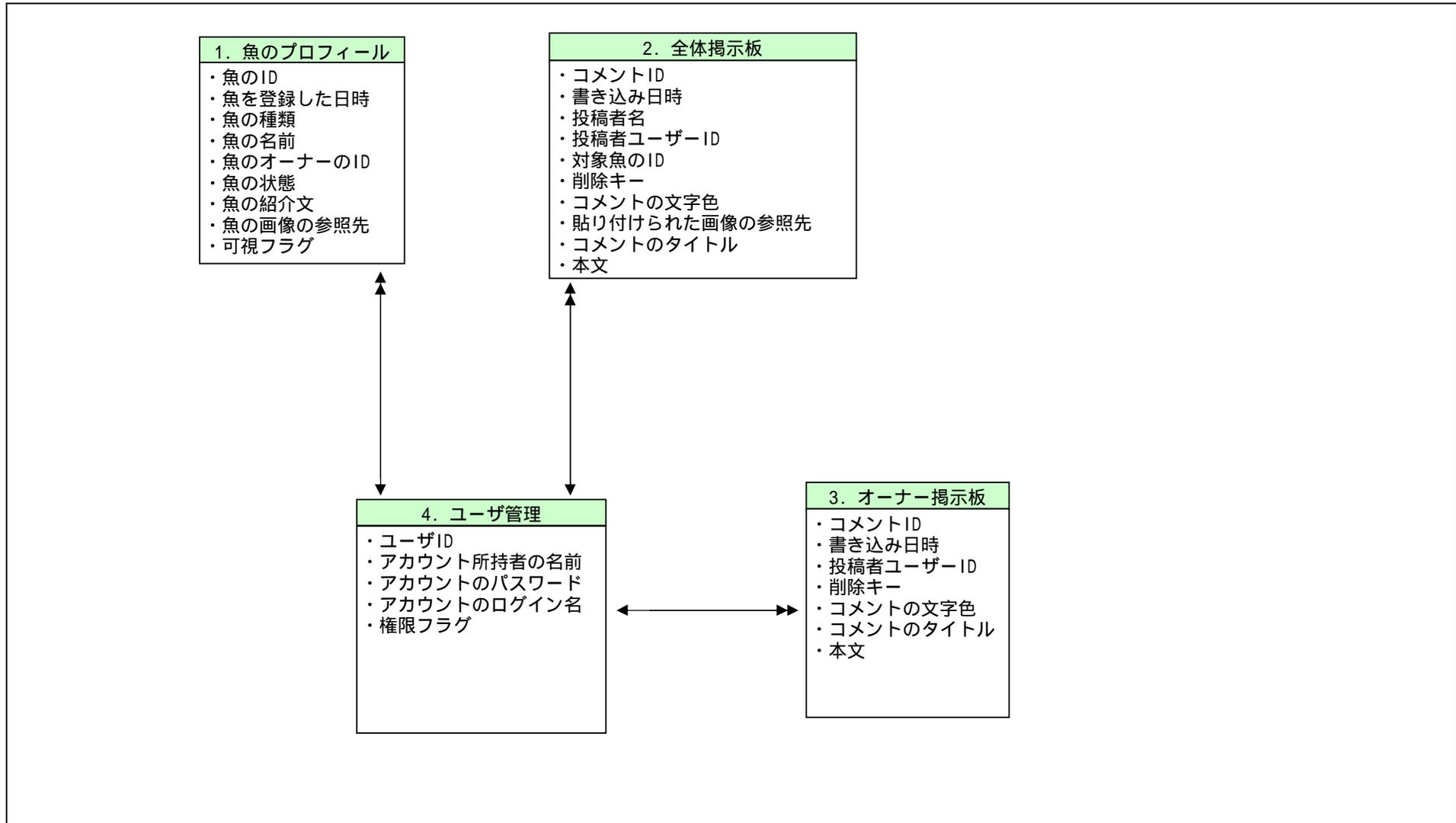
5 設計

5.1 データベースの設計

システムで取り扱うデータを格納するデータベースについて、基本設計を実施した。データを洗い出したところ、それぞれ以下のいずれかに分類することができると考えられたので、これらについてテーブル化することとした。

- ・ 人の情報
- ・ 魚の情報
- ・ 全体掲示板に関する情報
- ・ オーナー専用掲示板に関する情報

【成果物：テーブル関連図、テーブル仕様書】



1. 魚のプロフィール (魚の情報を保有するテーブル)

テーブル名

連番	p_key	項目名	カラム名	タイプ	レングス	内容	入力者	参照先マスター
1	*	魚のID(登録順)	fish_id	int	3	魚を一意に識別するID、登録した順に連番でつけられる。	オートインクリメント	
2		魚を登録した日時(西暦・月・日)	enrolled_date	DATE	-	魚の登録された日時	自動	
3		魚の種類	fish_type	TEXT	30	魚の種類の名称	オーナー	
4		魚の名前	fish_name	TEXT	50	魚の名前	オーナー	
5		魚のオーナーのID	owner_id	int	3	魚のオーナーのID	管理者	ユーザ管理TBL
6		魚の状態(生死)	fish_status	int	1	0-生きてる 1-行方不明 2-死亡	オーナー	
7		魚の紹介文	introduction	TEXT	300	魚を紹介する為の紹介文	オーナー	
8		魚の画像の参照先	pic_adress	TEXT	100	魚の画像が格納されているアドレス	自動	
9		可視フラグ	visible	int	1	可視 不可視を切り替える 0-不可視 1-可視 デフォルト: 0	自動 管理者	

2. 全体掲示板 (掲示板の書き込み情報を保有するテーブル)

テーブル名

連番	p_key	項目名	カラム名	タイプ	レングス	内容	入力者	参照先マスター
1	*	コメンID	comment_id	int	7	書き込まれた順に連番	オートインクリメント	
2		書き込み日時	comment_posted_date	DATETIME	-	書き込まれた年月日時	自動	
3		投稿者名	poster_name	TEXT	20	投稿者の名前	全ユーザ	
4		投稿者ユーザID	poster_owner_id	int	3	投稿者がオーナー 管理者である場合、そのID	自動	ユーザ管理TBL
5		対象魚のID	fish_id	int	3	対象の魚が居る場合、その魚のID	オーナー	魚のプロフィールTBL
6		削除キー	delete_key	TEXT	4	書き込みを削除する時の認証用文字列 スコープ外により未使用	オーナー	
7		コメントの文字色	comment_color	int	2	コメントの文字の色	全ユーザ	
8		貼り付けられた画像の参照先	pic_adress	TEXT	30	コメントに貼り付けられた画像の格納されている参照先	自動	
9		コメントのタイトル	comment_title	TEXT	20	コメントのタイトル欄	全ユーザ	
10		本文	comment_body	TEXT	500	コメント本文	全ユーザ	

3. オーナー掲示板（オーナー掲示板の書き込み情報を保有するテーブル）

テーブル名

連番	p_key	項目名	カラム名	タイプ	レンジ	内容	入力者	参照先マスター
1	*	コメントID	comment_id	int	7	書き込まれた順に連番	オートインクリメント	
2		書き込み日時	comment_posted_date	DATETIME	-	書き込まれた年月日時間	自動	
3		投稿者ユーザーID	poster_owner_id	int	3	投稿したオーナー 管理者のID	自動	ユーザ管理TBL
4		削除キー	delete_key	TEXT	4	書き込みを削除する時の認証用文字列 スコープ外により未使用	オーナー/管理者	
5		コメントの文字色	comment_color	int	2	コメントの文字の色	オーナー/管理者	
6		コメントのタイトル	comment_title	TEXT	20	コメントのタイトル欄	オーナー/管理者	
7		本文	comment_body	TEXT	500	コメント本文	オーナー/管理者	

4. ユーザ管理TBL（ユーザの管理情報を保有するテーブル）

テーブル名

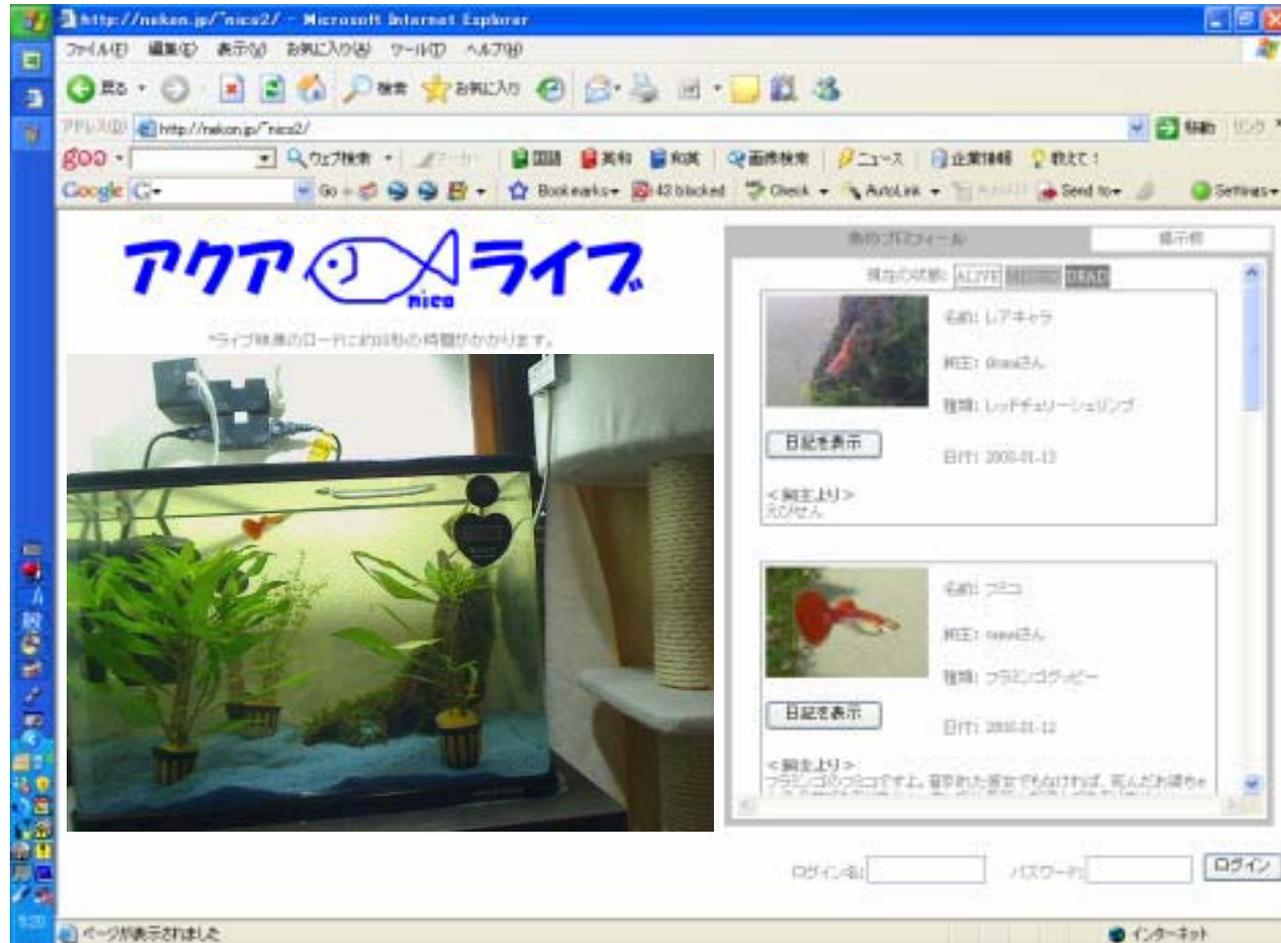
連番	p_key	項目名	カラム名	タイプ	レンジ	内容	入力者	参照先マスター
1	*	ユーザのID(登録順)	user_id	int	3	ユーザを一意に識別するID、登録した順に連番でつけられる。	オートインクリメント	
2		アカウント所持者の名前	name	TEXT	20	オーナーの名前	管理者	
3		アカウントのlogin名	login_name	TEXT	8	ログイン時に使用するログイン名	管理者	
4		アカウントのパスワード	login_password	TEXT	16	ログイン時に使用するパスワード	管理者	
5		権限フラグ	privs	int	1	0-オーナー 1-管理者	管理者	

5.2 画面の設計

クライアントへのヒアリングの際に用いたモックアップ・オブジェクトを土台に、画面の設計を実施した。画面のレイアウト、ならびに画面に表示・配置する項目についてドキュメント化し、メンバー間で共有した。

【成果物：画面説明】

画面名 (初期画面)



5.3 個別関数の定義

PHP 言語を利用したコミュニケーション・アプリの作成に関しては、2 人のメンバーによる分業とした。主に UI 部分を担当するメンバーと、裏で動く関数部分を担当するメンバーにわけたことで、それぞれの責任範囲を明確に示すことができた。

2人がスムーズに分業を行えるように、メンバー全員で関数の設計を実施した。これにより、それぞれが自担当部分についてプログラミングのみならず、テストドライバとスタブを用いた単体テストまでを個別に実施することができた。

【成果物：関数定義書】

bbs : 全体掲示板の情報を取り扱うクラス

1	関数名	connect_db	DBに接続する
	引数	(なし)	
	戻り値	(なし)	
2	関数名	public_add_comment	コメントを書き込む
	引数	\$name	投稿者名
		\$title	題名
		\$message	本文
		\$fish_id	魚のID
		\$owner_id	オーナーのID
		\$letters_color	文字の色
	\$picture	画像の名前	
	戻り値	int - 1 成功 2 投稿者名が空 3 本文が空 4 題名が空	
3	関数名	public_get_comment	一ページに表示するコメントを取得する
	引数	\$from	取得対象となるコメントの範囲の始点
		\$size	取得するコメントの数
	戻り値	array[] コメントのリスト	
	特記事項	取得範囲内のコメントの数が\$size未満の場合は、ある分だけ取得する	
4	関数名	public_get_fish_comment	ある魚に対するコメントを取得する
	引数	\$fish_id	魚のID
		\$from	取得対象となるコメントの範囲の始点
		\$size	取得するコメントの数
	戻り値	array[] コメント	
	特記事項	取得範囲内のコメントの数が\$size未満の場合は、ある分だけ取得する	
5	関数名	public_get_comment_count	コメントの総数を取得する
	引数	(なし)	
	戻り値	int コメントの総数	
6	関数名	public_get_fish_comment_count	ある魚に対するコメントの総数を取得する
	引数	\$fish_id	魚のID
	戻り値	int コメントの総数	
7	関数名	public_get_fish_comment_count	コメントを削除する。(書き込みの内容を----に置き換える)
	引数	\$comment_id	コメントのID
	戻り値	int - 1 成功	

owner_bbs : オーナー掲示板の情報を取り扱うクラス

1	関数名	connect_db	DBに接続する
	引数	(なし)	
	戻り値	(なし)	

2	関数名	owner_add_comment	コメントを書き込む
	引数	\$title	題名
		\$message	本文
		\$owner_id	オーナーのID
		\$letters_color	文字の色
	戻り値	int - 1 成功 2 本文が空 3 題名が空	

3	関数名	owner_get_comment	一ページに表示するコメントを取得する
	引数	\$from	取得対象となるコメントの範囲の始点
		\$size	取得するコメントの数
	戻り値	array[] コメントのリスト	
	特記事項	取得範囲内のコメントの数が\$size未満の場合は、ある分だけ取得する	

4	関数名	owner_get_comment_count	コメントの総数を取得する
	引数	(なし)	
	戻り値	int コメントの総数	

user : ユーザーの情報を取り扱うクラス

1	関数名	connect_db	DBに接続する
	引数	(なし)	
	戻り値	(なし)	

2	関数名	add_owner	オーナーを登録する
	引数	\$name	オーナーの名前
		\$login_name	ログイン名
		\$login_passwd	パスワード
	戻り値	int - 1 成功 2 オーナー名のエラー 3 ログイン名のエラー 4 パスワードのエラー	空/おかしい 空/おかしい/重複している 空/おかしい

3	関数名	login	ログイン認証をする
	引数	\$login_name	ログイン名
		\$login_passwd	パスワード
	戻り値	int - 1 管理者ログイン 2 オーナーログイン 3 パスワード誤り 4 ログイン名誤り	

4	関数名	get_owners_name	user_id からオーナーの名前を取得する
	引数	\$user_id	ユーザーのID
	戻り値	text オーナーの名前	ログイン名が存在しないときは、falseを返す

5	関数名	get_owner_list	ユーザーの名前とIDを取得しリストにする
	引数	(なし)	
	戻り値	int ID text ユーザーの名前	

6	関数名	is_admin	ログイン名からオーナーか管理者かどうか判別する
	引数	\$login_name	ログイン名
	戻り値	int - 1 オーナーである 2 管理者である 3 存在しないログイン名	

7	関数名	get_owners_id	ログイン名からユーザーIDを取得する
	引数	\$login_name	ログイン名
	戻り値	int ID	ユーザー名が存在しないときは、0を返す

fish : 魚の情報を取り扱うクラス

1	関数名	connect_db	DBに接続する
	引数	(なし)	
	戻り値	(なし)	
2	関数名	regist_fish	魚を登録する
	引数	\$owner_id	オーナーのID
		\$fish_name	魚の名前
		\$fish_type	魚の種類
		\$enrolled_date	登録日時
		\$introduction	紹介文
		\$picture	画像の名前
	戻り値	boolean - TRUE FALSE	成功 一致したデータが存在しない
3	関数名	modify_fish	魚を更新をする
	引数	\$owner_id	オーナーのID
		\$fish_id	魚のID
		\$fish_name	魚の名前
		\$fish_type	魚の種類
		\$enrolled_date	登録日時
		\$fish_status	魚の状態
		\$introduction	紹介文
		\$picture	画像の名前
		\$validation	有効/無効区分
	戻り値	boolean - TRUE FALSE	成功 オーナーじゃない/魚の名前・魚の種類・魚の登録日時が未入力
4	関数名	get_fish_count	登録されている(かつ公開されている)魚の数を取得する
	引数	(なし)	
	戻り値	int	登録されている魚の数
5	関数名	get_max_fish_id	最新の魚のIDを取得する(これから登録する魚は戻り値+1のIDに)
	引数	(なし)	
	戻り値	int	最新の魚のID
	特記事項		オーナーが魚を持っていない場合には、falseを返す 有効な魚の情報だけを取得する
6	関数名	get_owners_fish	オーナーIDから魚の情報を取得する
	引数	\$owner_id	オーナーのID
	戻り値	array[]	魚の情報 該当のfish_idが存在しないときは、falseを返す
7	関数名	get_fish	魚のIDから魚の情報を取得する
	引数	\$fish_id	魚のID
	戻り値	array[]	魚の情報 該当のfish_idが存在しないときは、falseを返す
8	関数名	get_fish_pager	指定の範囲の魚の情報を取得する
	引数	\$from	取得対象となる魚の情報の範囲の始点
		\$size	取得する魚の情報の数
	戻り値	array[]	魚の情報のリスト 始点がそもそもデータの範囲外の場合、falseを返す
	特記事項		データが指定された個数分無いとき、ある分だけ取得する

6 映像技術について

本プロジェクトの目的の1つであり、システムの肝とも言える「ライブ映像をリアルタイム配信する」という部分については、プロジェクトの初期段階から中盤以降まで、ずっと悩まされてきた。クライアントからの大きな要望により「Linuxサーバを用いた映像配信（Linuxで一本化）」に精力を注いできたが、結果としてそれを実現することはできなかった。

映像配信部分については、1人のメンバーが設計から実装までを一括して担当した。実際にLinuxサーバとWebカメラを用意し、数々の技術について検証を実施したが、そのたびに問題が発生し、方針変更を余儀なくされた。また、担当者が実施した作業の詳細について、PMを含めた他メンバーがきちんと把握していなかったことで、この部分に対する問題意識が薄く、最後まで担当者に負担をかけることになってしまった。

PMOからアドバイスをいただき、不具合について1つ1つの事象を追っていったところ、最終的には採用しようとしている技術に不具合があることが判明した。プロジェクトメンバーでどうにかできる問題ではなかったため、この時点でコンテンツエンジンプランを発動し、ストリーミング用にWindowsサーバを立てることとした。

クライアントの希望する「トータルLinux」については、技術が追いついてくれば実現可能となる。つまり、時間が解決するものであると考える。

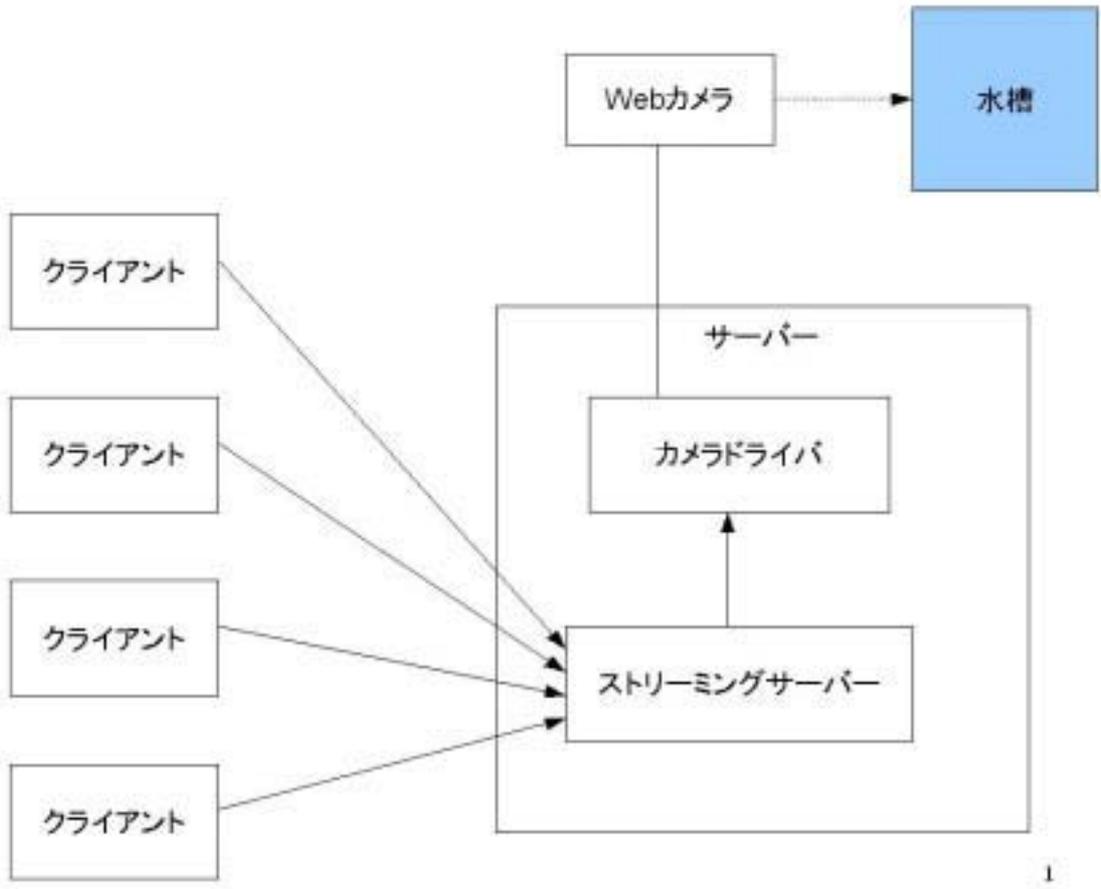
【成果物：映像技術に関する報告書】

映像技術に関する報告書

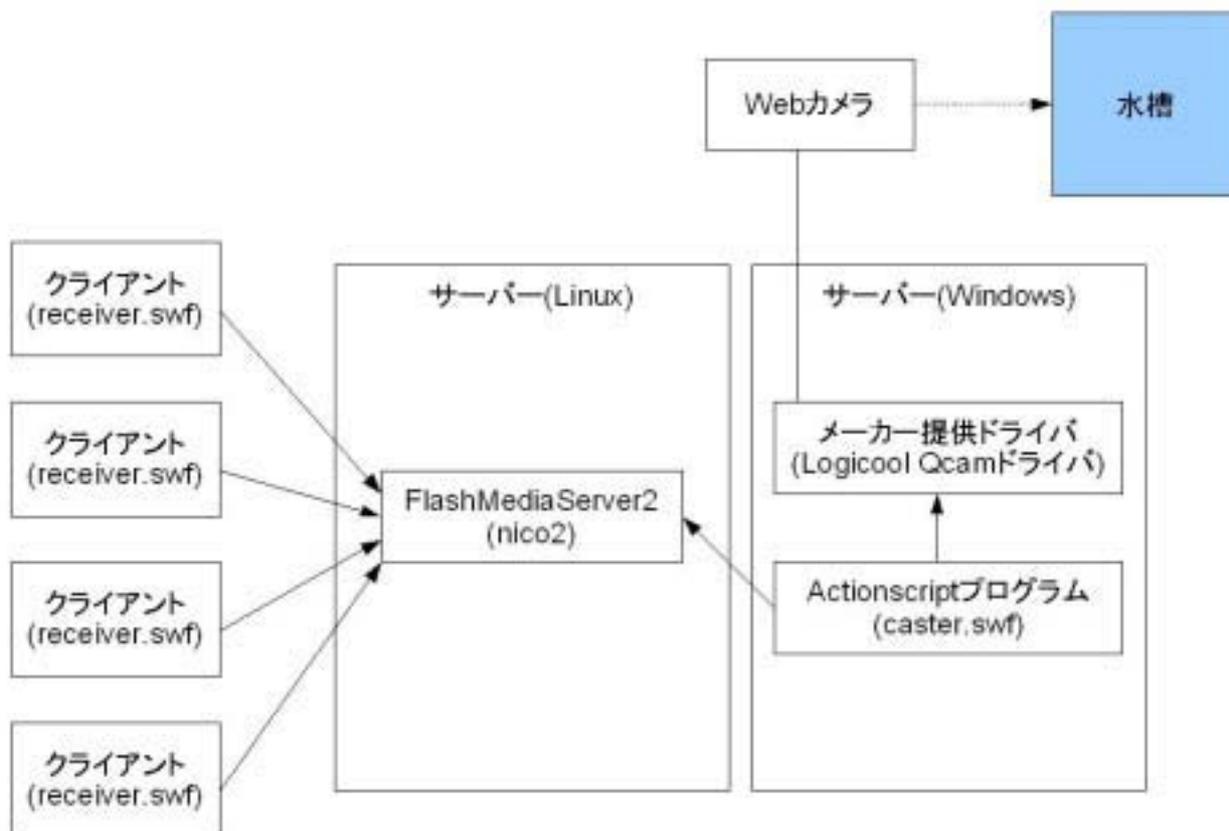
niconico ライブ
川口 将司

●カメラ報告書

以下の様なシステムの実現を目指した。



1. 実現したシステムの構成

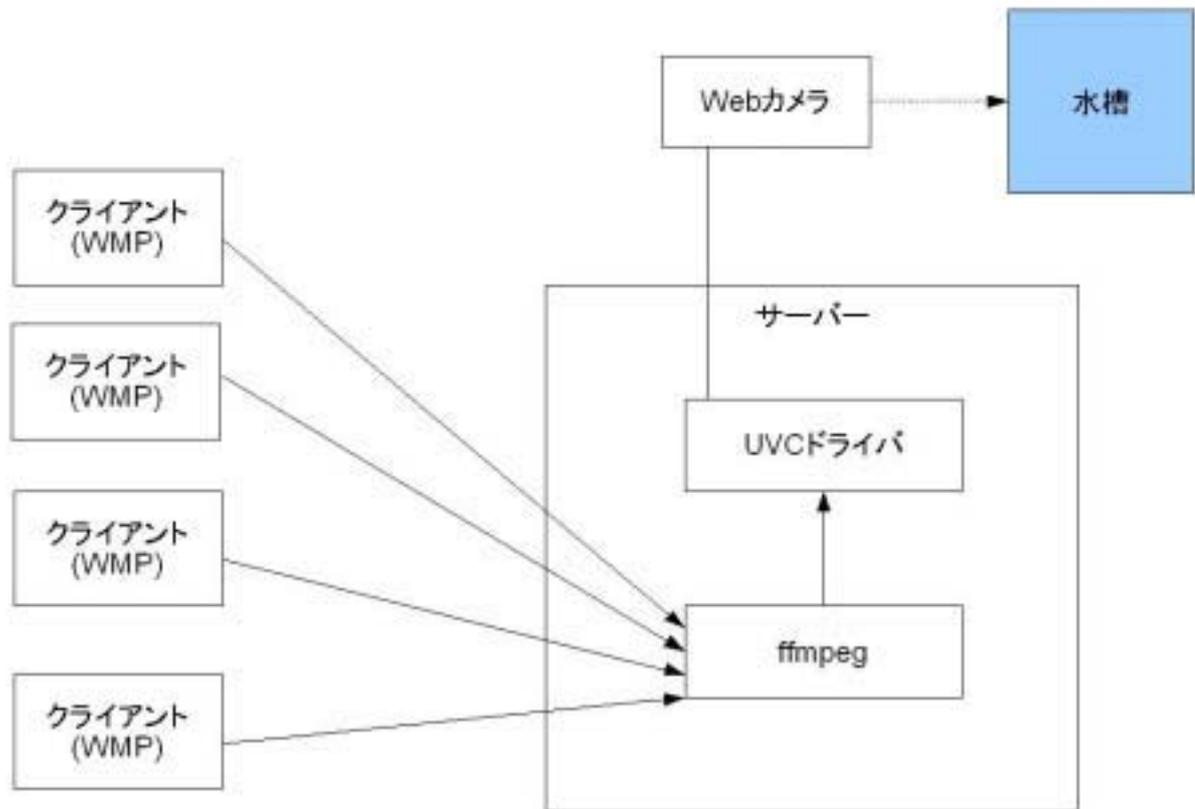


2. 実現出来なかった構成

当初、Linux マシンにカメラを接続して映像を取得（以下カメラ機能）し、ストリーミングサーバーの設置（以下ストリーミング機能）を行おうと考えていた。これに当たって二通りのシステム構成パターンがあった。

前提としているLinuxのOSはCentOS 5.0（アップデートを繰り返してる為、構成はほぼCentOS 5.1相当）。カメラドライバはUVC（USB Video Class）である。

・パターン1、FFmpeg



2

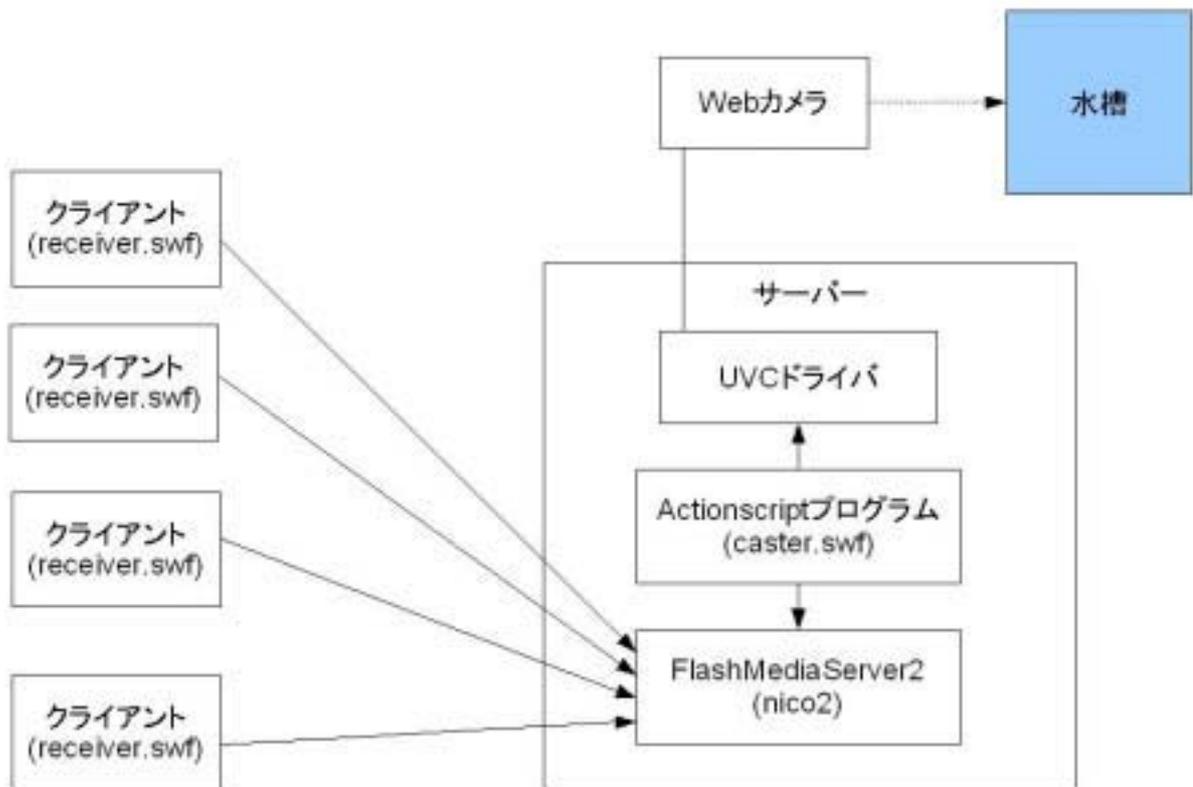
最初の案として、ストリーミングサーバーにFFmpegというオープンソースを利用しようと考えていた。結果から言うと成功しなかった。

実行の際、プログラムへ引数としてカメラを指定するのだが、指定しても正常に動作しなかった。後述する様に、カメラとカメラのドライバ自体には問題がなかったので、FFmpeg側のバグだったかと思われる。

具体的には、実行の際『引数を一つ以上与えて下さい』というエラーメッセージが出るのだが、引数を与えたところでエラーメッセージが変化しないものであった。

UVCドライバに問題があるのではないかと考えたが、Luvcviewというカメラから映像を取得してディスプレイに映すソフトウェアは動作したので、UVCに問題がある可能性はほぼ無くなった。

・パターン2、FlashMediaServer2 (Linux版)



3

代替案として、Adobe製品であるFlashMediaServer2 (以下FMS2) を利用する案があった。ただし無償のDeveloper Editionである。結果から言うところにも成功しなかった。

Linux版FMS2のサポート対象としているOSがRed Hatエンタープライズに限っていたが、CentOS5.0並びにFedora7でも動作は問題なかった。しかし別のところに問題があった。

構成図から見て取れる様に、カメラから映像を取得するプログラムはFlash/ActionScriptで書かれており、これの実行系としてFlashPlayerを利用することになる。しかしLinux版FlashPlayerではカメラから映像を取得するプログラムは正常に動作しなかった。Windows版FlashPlayerでは正常に動作するプログラムにも関わらず。

このことをAdobeの技術サポートセンターへ問い合わせたところ、Linux版FlashPlayerでは未だカメラから映像を取得する機能を実装していないとのことであった。Windows及びMacintosh版FlashPlayerのみで実装されている機能らしい。Linux版で当機能が実装される見通しは立っていないが、放置されたままということはないだろう、という係りの人の意見だった。

Adobeによって当機能さえ実装されれば、当案は実現可能である。もしくはAdobeが独自で用意しているRTMPと呼ばれるプロトコルを理解し、カメラ機能で得られた

映像をFMS 2へRTMPプロトコルで転送出来る様なプログラムを組めば、実現可能かと思われる。後者は納期と技術力の関係で達成出来なかった。

蛇足だが、最初Adobeのカスタマーサポートセンター(?)へ電話したのだが、Flash Playerは無償で配布しているソフトウェアである為、技術的な質問には一切お答え出来ないという回答を頂いた。技術サポートセンターでは答えて貰えたので、全くいい加減な対応だと思った。もしくはFMS 2という有償ソフトウェアの名前を出したから答えて貰えたのだろうか。

Linuxマシン一台でカメラ機能、ストリーミング機能を完結させようとしたが、以上の様な理由で実現出来なかった。そこでパターン2で問題とされていたFlash Playerの問題を、Windowsマシンに移行してしまうことで解決することにした。これが最終的に実現したシステムの構成である(1.を参照)。

3. デプロイ手順

(1) Linuxのみで配信する場合の手順

Flash Playerの機能の関係上、こちらの方法は実現不可。Flash Playerにカメラ機能が実装された場合、この方法で実現可能なはず。

(注) 先頭に\$や#が付いてるところはターミナル上で行うコマンドです。

ストリーミングサーバーの作成
と

Flashアプリケーション周りの設定
の二部編成です。

●まずはストリーミングサーバーを作成します。

【ここからは授業時に作業済みです】

rootユーザー(もしくは権限が同等のユーザー)になる。

まずはlibstdc++.so.5をインストール。

```
# yum install libstdc++.so.5
```

次にFMS2(Linux版)をここからダウンロードする。

```
http://www.adobe.com/cfusion/tdrc/index.cfm?loc=ja&product=flashmediaserver
```

ダウンロードしたディレクトリへ行き、

```
# tar zxvf FlashMediaServer2.tar.gz
```

出来たディレクトリをカレントにし、

```
# ./installFMS -platformWarnOnly
【ここまでは授業時に作業済みです】
```

libcrypto.so.*および libssl.so.*を/usr/lib 直下からシンボリックリンクで参照する。

```
# cd /usr/lib
# ln -s libcrypto.so libcrypto.so.4
# ln -s libcrypto.so libcrypto.so.6
# ln -s /lib/libssl.so.4 libssl.so.4
# ln -s /lib/libssl.so.6 libssl.so.6
```

【libssl.so.*が無い場合】

```
# yum whatprovides libssl.so.4
または
# yum whatprovides libssl.so.6
でリストされたパッケージを yum install する。
```

【libssl.so.*が無い場合 終わり】

FMS を起動する。

```
# chkconfig fms on
# service fms start
```

プロセスを確認。

```
# ps ax |grep fms
```

【fmsadmin が起動していない場合】

```
# cd /opt/macromedia/fms
# ./fmsadmin -console -conf ./conf/Server.xml
【fmsadmin が起動していない場合 終わり】
```

telnet でサーバー起動をテスト。

```
# telnet 127.0.0.1 1111
# telnet 127.0.0.1 1935
```

それぞれ Connected to 127.0.0.1...等と出れば OK。

火壁を利用している場合、iptables を編集して restart する。

開放が必要なポートはストリーミングサーバー用の"1935"と、遠隔設定コンソール用の"1111"。

後者は今回は取り合えずは必要ない。(ただし運用を行っていく上で利便性が上がる)

ただこの辺に関して、nekon 鯖は火壁使っていないのでおそらく大丈夫。

FMS 上にサーバーアプリケーションを新しく作成。

```
# cd /opt/macromedia/fms/applications
# mkdir nico2
これで rtmp://鯖 IP/nico2 が利用出来る様になる。
```

●次に、カメラから映像を拾って来て、ストリーミングサーバーへ送り、クライアントで受信するまでを作成します。

ユーザーは誰でも良いが、自分は一般ユーザーで実験した。

以下から配信用アプリケーション一式を拾って来る。

```
http://web.sfc.keio.ac.jp/~t04263mk/2007autumn/nico2/video_caster.zip
http://web.sfc.keio.ac.jp/~t04263mk/2007autumn/nico2/video_receiver.zip
```

解凍する。

```
$ gzip ./video_caster.zip
$ gzip ./video_receiver.zip
```

設定ファイルを編集する。

```
$ vi video_caster/config.xml
$ vi video_receiver/config.xml
```

<element id="rtmp"><element>の間を[rtmp://IP またはドメイン/サーバーアプリケーション]とする。

サーバーアプリケーションの部分は/opt/macromedia/fms/applications/直下に作ったディレクトリ名である。

以下は例である。

```
<element id="rtmp">rtmp://nekon.jp/nico2<element>
```

【ここからは GUI でしか試してません。X Window システムでログインして下さい。】
video_caster に含まれる caster.html をブラウザで開く。

選択を求められるものが出てきたら全て『許可する(または allow)』を選択。

Flash が埋め込まれてる領域にカーソルを合わせて右クリックする。
出て来た窓の中の『設定』または『settings...』をクリック。

一番右のタブをクリックして、接続している Web カメラが認識されているか確認する。

次にカメラのタブをクリックして高度な設定 (または advanced) をクリックする。
開かれた Web ページの左側のメニューにある Global Security Settings Panel をクリックする。

真ん中右辺りにあるタブの中から『追加する』を選択。
『参照』を押して caster.swf 及び receiver.swf を指定する。

両方とも『常に許可』に設定。

caster.html を再度読み込む。

カメラから映像を取得出来たら Flash の設定は成功している。

次に receiver.swf を起動。

カメラから映像を取得出来たら全工程は成功している。

(2) Windows + Linux で配信する場合の手順

Windows サイドの設定手順。Linux サイドは FMS 2 を入れるのみで、(1) と同じ手順。

Web カメラを Windows マシンに接続します。

ドライバはインストールしておいて下さい (購入時に付随していた奴で OK です)。

以下から配信用・受信用のアプリケーションを拾ってきます。

http://web.sfc.keio.ac.jp/~t04263mk/2007autumn/nico2/video_caster.zip

http://web.sfc.keio.ac.jp/~t04263mk/2007autumn/nico2/video_receiver.zip

適当なディレクトリに解凍して下さい。

config.xml を編集します。

今回は nekon.jp 上で動作している FMS の "nico2" というアプリケーションでストリーミング配信を行います。

<element id="rtmp">rtmp://nekon.jp/nico2<element> と編集して下さい。

別の鯖機で行う場合、nekon.jp の部分は適宜ドメインまたは IP に書き換えて下さい。

caster.html を起動します。

何やらポップアップで設定しろだの出てきましたら、ポップアップ中の『設定』ボタンを押して下さい。

出てこない場合は Flash が埋め込まれてる領域で右クリックをし、『設定』をクリックして下さい。

出て来たウィンドウの下方にある、一番左のタブを選択して下さい。

『高度な設定』を押して下さい。

以下の手順に従って設定をして下さい。

http://www.adobe.com/jp/devnet/flashplayer/articles/fp8_security_warning_02.html#05

caster.swf と receiver.swf 両方を指定して下さい。

設定が終わったら先ほど起動した Flash アプリケーションを再起動します。

ブラウザを一旦閉じて、再度 caster.html を開きます。

ポップアップが出てきたら『許可』を押して下さい。
とりあえずカメラが取得している映像が表示されたら成功です。

次に receiver.swf を開きます。

この時、receiver.swf のカレントディレクトリ中の config.xml も上と同様に設定して下さい。

カメラの映像が取得出来たら成功です。

7 テスト

PHP アプリの単体テストは、プログラム作成者による命令網羅のホワイトボックス方式のテストを採用することとした。前述のとおり、UI 部分の担当者と裏で動く関数部分の担当者を明確に分け、それぞれがテストドライバとスタブという形で互いのプログラムを補完し合うことで、個別に単体テストを実施することができた。

また、総合テストにおいては、事前に作成したテストケースを用いたブラックボックス方式のテストを採用することとした。

事前にメンバー全員で関数の設計を実施し、ドキュメント化していたにもかかわらず、プログラムの結合は一筋縄ではいかなかった。プログラムを作っていく中で、設計時点には思いつかなかった関数が必要になってきたとき、メール等でそれらの情報をやりとりしていたため、関数を呼ぶ側・呼ばれる側で引数や戻り値の仕様が異なってしまっていたことが原因である。また、ログイン情報の保持に利用される「`session`」オブジェクトや `image` ファイルを格納することができないという問題が発生したとき、その理由がどこにあるのか、切り分けを行うのに苦労した。実際は、関数云々の問題ではなく、単にファイルの書き込み権限の問題であったが。

一度結合に成功してからは、テストケースを用いてスムーズにテストを実施することができた。当初、セキュリティに対する考慮が欠けていたが、有識者からの指摘もあり、Web アプリならではのセキュリティ関連テスト等も実施し、イタズラ対策が実現できた。

【成果物：テストケース（結果記入済） 指摘事項一覧】

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	映像配信	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0020-1	シナリオ名称	映像画面の表示
--------	--------	--------	---------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
1	本システムを起動(Webサイトを表示)する	フレームに区切られた画面左側に映像配信枠があるか		12/26	1/29	
		映像が正常に表示されているか		12/26	1/29	
		フレームで区切られた画面右側の画面遷移に左右されないか		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	ログイン	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0030-1	シナリオ名称	オーナーの本システムへのログイン
--------	--------	--------	------------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	オーナー権限のアカウントのログイン名とパスワード					
1	ログインする					
2	ログイン名を記入する	ログイン名欄があり ログイン名を入力できること		12/26	1/29	
3	パスワードを記入する	パスワード欄があり パスワードを入力すると表示されること		12/26	1/29	
4	ログインボタンを押す	画面がリロードされること		12/26	1/29	
5	ログインが正常に完了したのを確認する					
6	ログイン後の画面であるか確認する	画面右上のタブに、オーナーツールタブが追加されていること		12/26	1/29	
7	全体掲示板タブを選択する	名前フォームが消え、画像参照・魚の選択リスト追加された状態であること		12/26	1/29	
8	ログイン情報がセッション管理されているか確認する	他のページが遷移/ページを更新してもログイン状態が保持されている		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	ログイン	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0030-2	シナリオ名称	管理者の本システムへのログイン
--------	--------	--------	-----------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	管理者権限のアカウントのログイン名とパスワード					
1	ログインする					
2	ログイン名を記入する	ログイン名欄があり ログイン名を入力できること		12/26	1/29	
3	パスワードを記入する	パスワード欄があり パスワードを入力すると表示されること		12/26	1/29	
4	ログインボタンを押す	画面がリロードされること		12/26	1/29	
5	ログインが正常に完了したのを確認する					
6	ログイン後の画面であるか確認する	画面右上のタブに、管理者ツールタブが追加されていること		12/26	1/29	
7	全体掲示板タブを選択する	名前フォームが消え、画像参照・魚の選択リスト追加された状態であること		12/26	1/29	
8	ログイン情報がセッション管理されているか確認する	他のページから遷移/ページを更新してもログイン状態が保持されていること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	ログイン	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0030-3	シナリオ名称	誤った情報でのログイン試行
--------	--------	--------	---------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	いずれかが間違った(空白含む)ログイン名とパスワード					
1	ログインする					
2	間違ったログイン名を記入する	ログイン名欄があり ログイン名を入力できること		12/26	1/29	
3	間違ったパスワードを記入する	パスワード欄があり パスワードを入力するとで表示されること		12/26	1/29	
4	ログインボタンを押す	ログインが失敗した旨のメッセージが表示されること	×	12/26	1/29	1/13 渡辺修正
5	ログインが正常に失敗したのを確認する	ログイン前の画面と変化無いこと		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	ログイン	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0030-4	シナリオ名称	オーナーのログアウト試行
--------	--------	--------	--------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	オーナーとしてログインした状態であること					
1	ログアウトする					
2	ログアウトボタンを押す	ボタンを押すと画面変化が起こること		12/26	1/29	
3	ログアウトが正常に行われたのを確認する					
4	ログアウト後の画面であるか確認する	オーナーツールタブが消え、ログインフォームが再出現すること		12/26	1/29	
5	全体掲示板タブを選択する	名前フォームが再出現、画像参照 魚の選択リストが消失すること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	ログイン	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0030-5	シナリオ名称	管理者のログアウト試行
--------	--------	--------	-------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	管理者としてログインした状態であること					
1	ログアウトする					
2	ログアウトボタンを押す	ボタンを押すと画面変化が起こること		12/26	1/29	
3	ログアウトが正常に行われたのを確認する					
4	ログアウト後の画面であるか確認する	管理者ツールタブが消え、ログインフォームが再出現すること		12/26	1/29	
5	全体掲示板タブを選択する	名前フォームが再出現、画像参照 魚の選択リストが消失すること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	魚のプロフィール	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0040-1	シナリオ名称	魚のプロフィール一覧の閲覧
--------	--------	--------	---------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
1	魚のプロフィールタブを選択	魚のプロフィール一覧画面が表示されること		12/26	1/29	
		登録されている全ての有効な魚のプロフィールが表示されること		12/26	1/29	
		名前欄にそれぞれの魚に応じた名前が表示されていること		12/26	1/29	
		飼い主欄にそれぞれの魚に応じたオーナーの名前が表示されていること		12/26	1/29	
		種類欄にそれぞれの魚に応じた種類が表示されていること		12/26	1/29	
		日付欄にそれぞれの魚に応じた登録日が表示されていること		12/26	1/29	
		飼い主より欄にそれぞれの魚に応じた魚についての情報が表示されていること		12/26	1/29	
		登録された魚の画像が適切に表示されること		12/26	1/29	
		画像がない場合はNO IMAGEと表示されること		12/26	1/29	
		それぞれの魚へのコメントを表示する為のボタンが表示されていること		12/26	1/29	
		背景がそれぞれの魚の状態に応じた色になっていること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	魚のプロフィール	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0040-2	シナリオ名称	魚のプロフィールの追加
--------	--------	--------	-------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	管理者アカウントでログイン済みであること					
1	管理者ツールタブを選択	管理者ツール画面が表示されること		12/26	1/29	
2	魚のプロフィールを追加する					
3	魚のprofileを追加を選択	魚のプロフィール追加画面へ遷移すること		12/26	1/29	
	飼主リストからオーナーを選択する	プルダウンの飼主リストが存在し 選択できること	×	12/26	1/29	1/16 鳴澤修正
		全ての登録されたオーナー 管理者の名前がリストアップされていること		12/26	1/29	
4	名前欄に魚の名前を記入する	名前欄が存在し 入力できること		12/26	1/29	
5	種類欄に魚の種類を記入する	種類欄が存在し 入力できること		12/26	1/29	
6	日付欄に魚の登録日時を記入する	日付欄が存在し 入力できること		12/26	1/29	
7	その他欄にその他魚についての情報を記入する	その他欄が存在し 入力できること		12/26	1/29	
8	画像欄右、参照ボタンより魚の画像を選択する	画像ファイルを選択するウィンドウによりファイルを選択できること		12/26	1/29	
		画像を参照しなくても追加が可能であること	×	12/26	1/29	1/13 渡辺修正
9	Profileを追加 ボタンを押し追加を完了する	追加が完了したゆしのメッセージが表示されること		12/26	1/29	
10	魚のprofileが正常に追加されたことを確認する					
11	魚のプロフィールタブを選択する	追加したプロフィールの内容が正しいこと		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	魚のプロフィール	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0040-3	シナリオ名称	魚のプロフィールの更新
--------	--------	--------	-------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	オーナー又は管理者アカウントでログイン済みであること					
1	オーナー(又は管理者)ツールタブを選択	オーナー (又は管理者)ツール画面が表示されること		12/26	1/29	
2	魚のプロフィールを更新する					
3	魚のprofileを更新を選択	更新可能な(所有している)魚のプロフィール一覧が表示されること		12/26	1/29	
4	名前欄の魚の名前を改変する	名前欄に変更前の名前が記入済みであり 変更できること		12/26	1/29	
5	種類欄の魚の種類を改変する	種類欄に変更前の種類が記入済みであり 変更できること		12/26	1/29	
6	日付欄の魚の登録日を改変する	日付欄に変更前の魚の登録年月日が記入済みであり 変更できること		12/26	1/29	
7	飼主より欄の情報を改変する	飼主より欄に変更前の情報が記入済みであり 変更できること		12/26	1/29	
8	画像欄右、参照ボタンより魚の画像を選択する	画像ファイルを選択するウインドウによりファイルを選択できること		12/26	1/29	
		画像を参照しなければ、以前の画像が引き継がれること		12/26	1/29	
9	魚の状態を選択する	ラジオボタンにより 魚の状態を選択することができる。		12/26	1/29	
10	公開/非公開を選択する	ラジオボタンにより公開/非公開を選択できる。		12/26	1/29	
11	Profileを更新」ボタンを押し更新を完了する					
	入力事項に不備がある場合	更新失敗のゆしのメッセージが表示されること		12/26	1/29	
	入力事項に不備がない場合	ページがリロードされること		12/26	1/29	
12	魚のprofileが正常に更新されたことを確認する					
13	魚のプロフィールタブを選択する	更新されたプロフィールの内容が正しいこと	×	12/26	1/29	1/29 渡辺修正

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	魚のプロフィール	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0040-4	シナリオ名称	特定の魚に対するコメントの閲覧
--------	--------	--------	-----------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
1	魚のプロフィールタブを選択	魚のプロフィール一覧が表示されること		12/26	1/29	
2	特定の魚に対するコメントを閲覧する					
3	対象魚のプロフィールの日記を表示ボタンを押す	対象魚に対するコメントの一覧が正常に表示されること		12/26	1/29	
		対象魚の名前が表示されていること		12/26	1/29	
		対象魚のオーナーの名前が表示されていること		12/26	1/29	
		対象魚へのコメントの貼り付け画像が正常に表示されていること	x	12/26	1/29	1/17 渡辺修正
		対象魚へのコメントの内容が正しいこと		12/26	1/29	
		コメントがない場合はそのゆしのメッセージが表示されていること		12/26	1/29	
4	ページングを切り替える	ページングの指定範囲に応じたコメントが表示されていること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	ユーザー管理	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0050-1	シナリオ名称	オーナーの追加
--------	--------	--------	---------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	管理者アカウントでログイン済みであること					
1	管理者ツールタブを選択	管理者ツール画面が表示されること		12/26	1/29	
2	オーナーを追加する					
3	オーナーの追加を選択	オーナーの追加画面が表示されること		12/26	1/29	
4	オーナー名欄にオーナーの名前を入力する	オーナー名入力欄があり 入力ができること		12/26	1/29	
5	ログイン名欄にログイン名*1を入力する	ログイン名入力欄があり 入力ができること		12/26	1/29	
6	パスワード欄にパスワードを入力する	パスワード入力欄があり 入力ができること		12/26	1/29	
7	「オーナーを登録」ボタンを押す	正常に登録され、そのゆしのメッセージが表示されること		12/26	1/29	
		入力事項がかけるとエラー表示がでること		12/26	1/29	
		ログイン名がすでに登録されているとそのメッセージが表示されること		12/26	1/29	
8	登録した内容でログインしてみる	登録した内容で正常にログインできること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	全体掲示板	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0060-1	シナリオ名称	コメントの閲覧
--------	--------	--------	---------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
1	全体掲示板タブを選択	これまでに投稿されたコメントの最新ページが表示されること		12/26	1/29	
		これまでのコメントが一定の件数ごとにページングされていること		12/26	1/29	10件ごと
		管理者としてログインしていなければ削除ボタンが表示されていないこと	×	12/26	1/29	1/13 渡辺修正
2	ページングを切り替える	ページングの指定範囲に応じたコメントが表示されていること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	全体掲示板	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0060-2	シナリオ名称	一般ユーザーによるコメントの書き込み
--------	--------	--------	--------------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
1	全体掲示板タブを選択	全体掲示板が表示されること		12/26	1/29	
2	コメントを書き込む					
3	名前欄に投稿者の名前を記入する	名前欄が存在し 入力できること オーナー 管理者と同じ名前を入力できること		12/26	1/29	
4	タイトル欄にコメントのタイトルを記入する	タイトル欄が存在し 入力できること		12/26	1/29	
5	内容欄にコメントの内容を記入する	内容欄が存在し 入力できること		12/26	1/29	
6	コメントの文字色を選択する	ラジオボックスで文字色が選択できること		12/26	1/29	
7	書き込み」ボタンを押し書き込みを完了する	ページがリロードされること		12/26	1/29	
8	コメントが正常に書き込まれたことを確認する	書き込まれた内容が正しく表示されていること		12/26	1/29	
		改行が行われていること	×	12/26	1/29	1/13 渡辺修正
		タグが使えなくなっていること	×	12/26	1/29	1/13 渡辺修正
		自動リンクがはられていること		12/26	1/29	
		”、’が文字として表示されること	×	12/26	1/29	1/13 渡辺修正

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	全体掲示板	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0060-3	シナリオ名称	オーナー 管理者によるコメントの書き込み
--------	--------	--------	----------------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	オーナー又は管理者アカウントでログイン済みであること					
1	全体掲示板タブを選択	全体掲示板が表示されること		12/26	1/29	
2	コメントを書き込む					
4	タイトル欄にコメントのタイトルを記入する	タイトル欄が存在し 入力できること		12/26	1/29	
5	内容欄にコメントの内容を記入する	内容欄が存在し 入力できること		12/26	1/29	
6	コメントの文字色を選択する	ラジオボックスで文字色が選択できること		12/26	1/29	
7	参照ボタンから貼り付ける画像を選択する	ファイルの選択ウィンドウが表示され画像を選択できること		12/26	1/29	
		画像を参照しなくても書き込みができること		12/26	1/29	
8	書き込みボタンを押す書き込みを完了する					
	書き込みに不備があるとき	書き込み失敗を表すメッセージが表示されること		12/26	1/29	
		書き込んだ情報が保持され残っていること		12/26	1/29	
	書き込みに不備がないとき	ページがリロードされること		12/26	1/29	
9	コメントが正常に書き込まれたことを確認する	書き込まれた内容が正しく表示されていること		12/26	1/29	
		画像が適切に表示されていること		12/26	1/29	
		名前が自動的に取得され表示されていること		12/26	1/29	
		改行が行われていること	×	12/26	1/29	1/13 渡辺修正
		タグが使えなくなっていること	×	12/26	1/29	1/13 渡辺修正
		自動リンクがはられていること		12/26	1/29	
		”、’が文字として表示されること	×	12/26	1/29	1/13 渡辺修正

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	全体掲示板	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0060-4	シナリオ名称	オーナー 管理者による魚に対するコメントの書き込み
--------	--------	--------	---------------------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	オーナー又は管理者アカウントでログイン済みであること					
1	全体掲示板タブを選択	全体掲示板が表示されること		12/26	1/29	
2	魚に対するコメントを書き込む					
3	プルダウンからコメント対象を魚を選択	プルダウンに所有している魚が表示され選択できること		12/26	1/29	
4	タイトル欄にコメントのタイトルを記入する	タイトル欄が存在し入力できること		12/26	1/29	
5	内容欄にコメントの内容を記入する	内容欄が存在し入力できること		12/26	1/29	
6	コメントの文字色を選択する	ラジオボックスで文字色が選択できること		12/26	1/29	
7	参照ボタンから貼り付ける画像を選択する	ファイルの選択ウィンドウが表示され画像を選択できること		12/26	1/29	
		画像を参照しなくても書き込みができること		12/26	1/29	
8	書き込みボタンを押して書き込みを完了する					
	書き込みに不備があるとき	書き込み失敗を表すメッセージが表示されること		12/26	1/29	
		書き込んだ情報が保持され残っていること		12/26	1/29	
	書き込みに不備がないとき	ページがリロードされること		12/26	1/29	
9	コメントが正常に書き込まれたことを確認する	書き込まれた内容が正しく表示されていること		12/26	1/29	
		名前が自動的に取得され表示されていること		12/26	1/29	
		画像がある場合は画像が正しく表示されていること		12/26	1/29	
		改行が行われていること	×	12/26	1/29	1/13 渡辺修正
		タグが使えなくなっていること	×	12/26	1/29	1/13 渡辺修正
		自動リンクがはられていること		12/26	1/29	
		”、’が文字として表示されること	×	12/26	1/29	1/13 渡辺修正

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	オーナー 管理者掲示板	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0070-1	シナリオ名称	コメントの閲覧
--------	--------	--------	---------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	オーナー又は管理者アカウントでログイン済みであること					
1	オーナー(管理者)ツールタブを選択	オーナー(管理者)ツール画面が表示されること		12/26	1/29	
2	オーナー 管理者掲示板を選択	オーナー 管理者掲示板画面が表示されること		12/26	1/29	
		これまでのコメントが一定の件数ごとにページングされていること		12/26	1/29	10件ごと
3	ページングを切り替える	ページングの指定範囲に応じたコメントが表示されていること		12/26	1/29	

テストシナリオ / ケース	プロジェクト	区分	起票	更新	承認
	niconicoライブ	オーナー 管理者掲示板	12/12	1/29	1/31
			鳴澤	渡辺	高橋

シナリオID	0070-2	シナリオ名称	オーナー 管理者によるコメントの書き込み
--------	--------	--------	----------------------

#	テストの流れ	確認内容	結果	確認	検証	備考
	【前提条件/準備】					
	本システムが起動(Webサイトが表示)していること					
	オーナー又は管理者アカウントでログイン済みであること					
1	オーナー(管理者)ツールタブを選択	オーナー(管理者)ツール画面が表示されること		12/26	1/29	
2	オーナー 管理者掲示板を選択	オーナー 管理者掲示板画面が表示されること		12/26	1/29	
3	コメントを書き込む					
4	タイトル欄にコメントのタイトルを記入する	タイトル欄が存在し 入力できること		12/26	1/29	
5	内容欄にコメントの内容を記入する	内容欄が存在し 入力できること		12/26	1/29	
6	コメントの文字色を選択する	ラジオボックスで文字色が選択できること		12/26	1/29	
7	書き込み ボタンを押し書き込みを完了する					
	入力事項に不備がある場合	書き込み失敗のゆしのメッセージが表示されること	×	12/26	1/29	1/13 渡辺修正
	入力事項に不備がない場合	ページがリロードされること		12/26	1/29	
8	コメントが正常に書き込まれたことを確認する	書き込まれた内容が正しく表示されていること		12/26	1/29	
		名前が自動的に取得され表示されていること		12/26	1/30	
		改行が行われていること	×	12/26	1/29	1/13 渡辺修正
		タグが使えなくなっていること	×	12/26	1/29	1/13 渡辺修正
		自動リンクがはられていること		12/26	2/2	
		”、 ’が文字として表示されること	×	12/26	1/29	1/13 渡辺修正

【iconicoライブ】指摘事項一覧

No	イマイチポイント	対応案	優先度	対応	
				日	者
1	入力項目に何か不備があったとき、エラーメッセージが表示され入力項目がクリアされてしまう	フォームの中身をクリアしないようにする。	高	1/13	渡辺
2	ログインはできるのに、ログアウトできない。	ログアウト機能を作成する。	中	1/13	渡辺
3	掲示板や魚プロフィールを登録したあと、その内容がすぐに画面に反映されない。 「正常に登録されました」という旨のメッセージは出るが、リロードしないと内容が反映されない。	すぐに反映されるように対応する。	中	1/13	渡辺
4	全体掲示板にイタズラ投稿された際に、管理者の方で当該文書を消すようなことができない。	管理者が投稿文書を管理できるようにする。 当初のスコープ外ですが、現実的に考えて必要ですね	低	1/13	渡辺
5	掲示板の「投稿日時」欄に、日付しか出ていない。	時間も出るようにする。	中	1/15	鳴澤
6	管理者が「魚のプロフィールの追加」をするとき、飼い主のログイン名をフォームに打ちこまなくては いけない。	リストから選択できるようにする。 リストとして出す項目はログイン名が良いのか？ オーナー名のほうが良くないだろうか？	中	1/16	渡辺
7	sessionからログイン情報を取り出すところや、DBへアクセスするところなど、共通の処理がさまざま なphpソースに書かれている。何か変更があったとき、メンテナンスが大変！ DBへアクセスする際の情報は3箇所書かれていました・・・ user.php / bbs.php / fish.php	共通phpを作り、そのソースをインクルードするようにする。	低	-	-
8	掲示板と魚のプロフィールのページングが不便。最新10(5)件を表示するようにしてほしい。	最新10(5)件を表示するようにする。	高	1/13	渡辺
9	掲示板の投稿の際、改行が無効になる。意識的に タグを入れないと改行されない。	改行コードを見るようにする。	中	1/13	渡辺
10	掲示板にリセットボタンはついていないほうがいい。書き込みボタンの隣にあるのは悲劇を生む。	リセットボタンをはずす。	高	1/13	渡辺
11	魚のプロフィールを更新したら写真が消えた。 その他項目を変えただけで、写真を指定していないと消える。	更新ロジックの見直し。	高	1/13	渡辺
12	日記の投稿をしたときに画像が正しくアップされなかった。	画像保存時のネーミングルール周辺を見直し。	高	1/13	渡辺
13	ログインをミスしたとき、何も(エラーメッセージの表示とか)出てこない。	ログインエラーの旨、表示する。	中	1/13	渡辺
14	未ログイン状態のとき、ログインアカウントと同じ名前でも「全体掲示板」に書き込むと、投稿者名が 空です」となる。	ログインアカウントと同じ名前でも書き込めるように対応する。 無理なら、せめてエラーメッセージをマッチさせる。	高	1/13	渡辺
15	掲示板の投稿に、SCRIPTタグを含むことができちゃう。イタズラのもとに？	無効化。	高	1/13	渡辺
16	掲示板の投稿を削除する権限のないユーザ(=管理者以外のユーザ)の画面にも、削除する「ボ タンがある。	ログインしているユーザが管理者の場合にのみ、ボタン表示。 押せないボタンは表示しないがユーザビリティの基本	高	1/13	渡辺
17	掲示板において「(ダブルコーテーション)」を文字として入力できず、この「」はソース内に埋め込 まれてしまう。悪意のあるユーザは、クロスサイトスクリプティングが可能に！	掲示板において入力された「」を文字として認識するように修正する(エスケープする?)。	高	1/13	渡辺
18	最初にライブ映像が配信されるまでに10秒ほど時間がかかる(これ自体は技術上の制約)。	ライブ映像のロードに約10秒ほど要する旨を、画面内(システム名の下あたり?)に書いておく。	中	1/15	渡辺
19	魚のプロフィールを登録/更新するとき、「その他」欄を空白にするとエラーとなる。入力必須項 目とするならば、「その他」という項目名は変えたほうがいい。	どの項目が必須項目なのかわかるようなUIにするのが理想。 例えば、項目名のうしろに(*)ってつけるとかもアリ。	中	1/16	渡辺 鳴澤
20	掲示板の投稿はカラフルでいいけど、「魚のプロフィール」や「魚の日記」がグレーで暗い？	「魚の日記」については、掲示板投稿の際に使用した色を引き継ぐとか。	低	1/15	渡辺
21	魚のプロフィールが、ちょっと見にくい感じがする。	「名前:」や「飼主:」などの項目名を右寄せにすると多少見やすいかもかもしれません。 <TD align="RIGHT">とか？ あと、魚と魚の間に水平線をひいた方が見やすい。	低	1/15	渡辺
22	Firefoxに対応しきれていない	要勉強	低	-	-
23	掲示板にリンクははれるようにしたい	check関数からhttp://をぬぐ行を削除。	中	1/13	渡辺
24	掲示板に現在のページ 前の10件後の10件表示	ページングルール見直し	中		渡辺
25	ログイン時に動画サーバがきれる	ログイン部分を右側に	高	1/15	渡辺
26	掲示板テーブルの寄せが違う	テーブル表示の見直し	低		渡辺
27	削除が一度につき1つしか行えない	削除のチェックボックス名をひとつずつ変え、配列にいれ、0かfalseでないものについて削除関数を実行するようにする	高	1/20	渡辺
28	ログイン情報が一定時間たつと消えてしまう？	セッション情報の破棄について勉強	低	-	-

8 クライアントからの評価

8.1 クライアントテストとアンケートへの回答

作成したシステムをクライアントに実際に使用していただき、提案書の内容を満たしているか、実稼動に耐えられるかについて確認していただいた。その上で、アンケートに思いのたけをぶつけていただいた。

結果は、概ね満足いくものだったものの、やはりトータル Linux での映像配信が実現できなかったこと、また学生メンバーからの連絡があまりなかったこと、これらに起因する評価が低く、全体平均では 5 段階のうち 3.8 という評価であった。

【成果物：松尾さんによるアンケート回答】

hiconico ライブプロジェクトに関するアンケート

【システムについて】

1. 当システムは、使いやすい作りになっていると感じますか？

強く思う まあまあ思う どちらとも言えない あまり思わない 思わない

コメント:直感的に操作できてよいと思います。あえて不便があるとしたら、オーナーに毎回魚の操作について教えないといけないところでマイナス1点。

2. 当システムにより、熱帯魚に関心のある人たちのコミュニケーションが活性化したと感じますか？

強く思う まあまあ思う どちらとも言えない あまり思わない 思わない

コメント:松澤さんや佐藤さんといった今まで飼ったことのない人に観てもらえたのは嬉しかったです。高橋さんも盛り上げありがとうございました。

3. 今回の開発範囲には満足していますか？

大変満足 まあまあ満足 普通 少し不満足 大変不満足

コメント:要求したとおりの物で満足の出来です。1月に入ってから細かい要求にも応じてもらえて使いやすい物に仕上がりました。

4. このシステムを使い続けたいと感じますか？

強く思う まあまあ思う どちらとも言えない あまり思わない 思わない

コメント:使いつづけるためには Linux 化が必須でした。当初の目的の友達の会社にも置けなくなってしまったので…。

5. その他、システムについてのコメントがあればお願いします。

基本要件には応えてもらったので、欲を言えば、プラスアルファの面白くなるアイデアが欲しかった？でも贅沢な要求です。

【プロジェクトについて】

1. niconico ライブのプロジェクトの進め方についてはどう感じましたか？

大変良い まあまあ良い 普通 あまり良くない 全然良くない

コメント:初期の段階でモックを作ってイメージのやりとりができたのがとても印象的でした。あれがなかったら意識のずれ違いが出来て大変だったに違いなし。

2. 一方的ではなく、きちんとコミュニケーションが取れたと感じますか？

強く思う まあまあ思う どちらとも言えない あまり思わない 思わない

コメント:学生メンバーからもっと連絡あってもよかったかな、と途中で思っていました。

3. その他、プロジェクトについてのコメントがあればお願いします。

学生3人ともそれぞれの範囲でしっかり動いてくれて、いいものに仕上がりました。PM の高橋さんまともめ役 + 進行ご苦労様でした。ではではー。

ご協力ありがとうございました。

8.2 今後の展望

メンバー間でプロジェクトの振り返りを実施する中で、今回作成したシステムの今後の展望について議論（フリートーク）した。

以下に、フリートークで挙げた内容を列挙する。

- ・画像（静止画）の保存 「カシャー」って鳴ると面白い
- ・動画が録画できて、それをみんなに見せられればなお面白い
- ・複数台のカメラの切り替え
- ・コメントやトラックバックの充実（魚の日記に対して）
- ・エサがやれる / 水槽をたたける
- ・水槽管理者に対してお金（ポイント）が渡せる、といった仕組み
- ・魚だけだとライブ映像っぽさが薄い（何かライブっぽいイベントとの両立）
- ・魚を買って～という依頼からお金の支払い等、本当のネット水槽商品っぽく

上記には、時間的あるいは技術的な制約から、今回のプロジェクトで削った機能も含まれている。最初に「こういうのがあると面白いのでは」と感じたものについて、プロジェクト終了後にも改めて同様の思いを得るということは、非常に興味深い。可能であれば、今後も継続して上記の機能を盛り込む開発を実施していきたいが、これについては、クライアントである松尾さんとよく相談することにする。

9 プロジェクト総括

今回のプロジェクトは、成功したと言って良いと考える。これは、最初にクライアントから提示された「ブラウザ経由での動画配信とコミュニケーションツールの作成」を実現することができたという一点に集約される。

クライアントからの要望として強かった「オール Linux での映像配信」は、技術的な制約から実現できなかったものの、どこに問題があったのか、どこが変われば実現可能になるのか、そのときの手順は、という詳細な報告レポートを記述し納品したことで、問題点はじゅうぶんにカバーできていると考える。

今回、つまりきながらもプロジェクトを完遂することができたのは、クライアントである松尾さんのお力添えに因るところが大きい。特に、映像配信部分について技術者としてのアドバイスをメンバーにしてくださったこと、ご自宅にあるサーバにテスト環境（のちの本番環境）を構築してくださったこと、細かいチューニングをしてくださったこと、Windows サーバを利用した映像配信部分のインターフェースを実装してくださったこと、その他あげていくときりがないほどたくさんのご協力をいただいた。ご本人がプロの技術者であることから、多くのクライアントにありがちな「無茶な要求」を投げかけてくるようなこともなく、逆に「これは難しいだろうな」という部分については細やかなフォローをしてくださった。本当に感謝の念が尽きない。

今回作ったシステムは、本当の意味で「クライアントと共に作ったシステム」であると言えるだろう。今後使い続けてもらうためには Linux 一本で完結させることが必須とのことだが、これについては時間が解決してくれるものだと考えている。

10 付録

毎週の進捗報告プレゼンテーション用資料、およびPMが作成した週報を付録として添付する。