

目次	
背景	3
目的	3
第1章 高齢化社会における在宅ケアサービス	4
1.1 高齢者医療の現状	4
1.2 問題点	8
第2章 医療情報化と高齢者用医療端末	10
2.1 ITの医療化	10
2.2 医療情報化	11
第3章 家庭における医療情報化	12
3.1 家庭における医療情報化	12
3.2 セルフメディケーション	14
第4章 家庭医療端末の情報化	15
第5章 解決へのアプローチ	17
第6章 設計と実装	18
6.1 設計	18
6.1.1 設計目標	18
6.1.2 構成要素と機能	19
6.1.3 システム構成	23
6.1.4 テーブルの設計	24
6.1.4 入力画面インタフェースの設計	25
6.2 実装	27
6.2.1 実装環境	27
6.2.3 機能の実装	29
第7章 評価	43
7.1 機能評価	43
7.2 考察	47
第8章 まとめと今後の課題	48
8.1 まとめ	48
8.2 今後の課題	48

参考文献	49
謝辭	50

背景と目的

背景

現在の高齢者医療・在宅ケアサービスには多種多様なものが多く存在するが、そもそも高齢者の自己管理能力が低下している。何かあれば病院へ、という考えがあるのは人間誰しもあることだが、まず自分の力で体調を管理することが大切である。体調を管理するためには、記録をとることがまず必要である。

目的

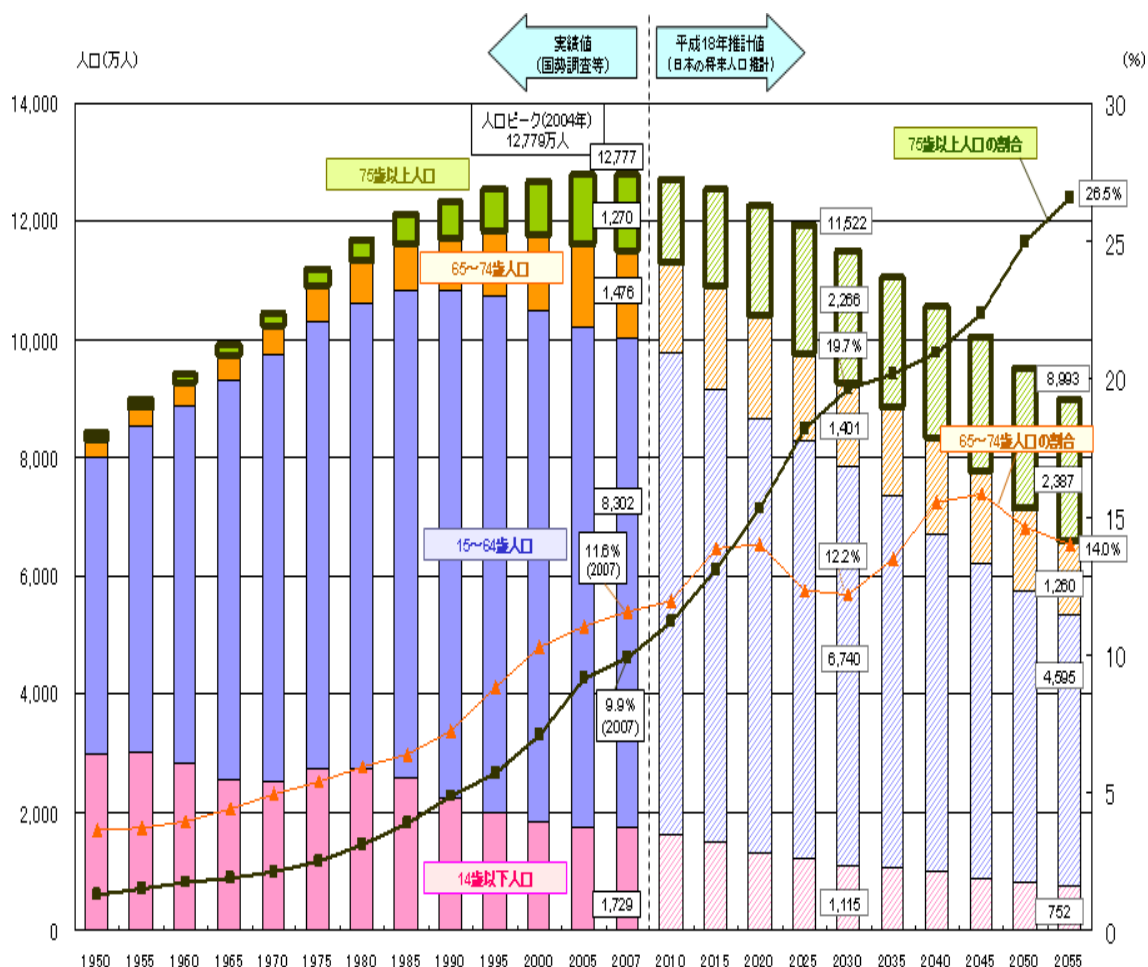
この研究では、個人の健康状態をデータベースに記録し、医療機関、家族に利用者の健康状態を送信することができ、その個人の健康状態をいつでも把握できるようなシステムを検討する。

1 章 高齢化社会における在宅ケアサービス

この章では、高齢者の現状を医療・在宅ケアサービスの分野からまとめ、問題点を述べる。1.1 では高齢者医療の現状を述べる。1.2 では現状からの問題点を述べる。

1.1 高齢者医療の現状

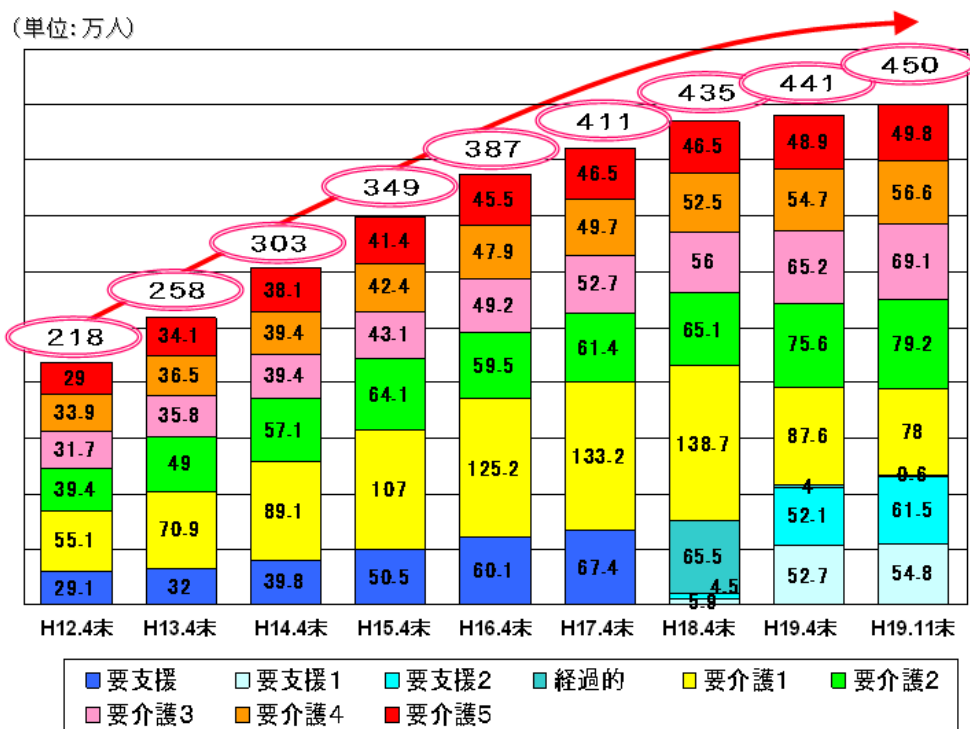
2009 年、我が国の 65 歳以上の高齢者の人口は 4 万 2 千 359 人である。総人口 1 億 2769 万 2 千人の 2 割を超え、今後さらに高齢化が進展すると予測されている（図 1）。



(出典)2006年までは総務省統計局「国勢調査」、2007年は総務省統計局「推計人口(年齢)」、2010年以降は国立社会保障・人口問題研究所「日本の将来推計人口(平成18年12月推計)中位推計」

図 1.1 人口の将来推計

また、高齢者人口の増加とともに、要介護認定を受けた高齢者の数も年々増加している
(図 1.2)。

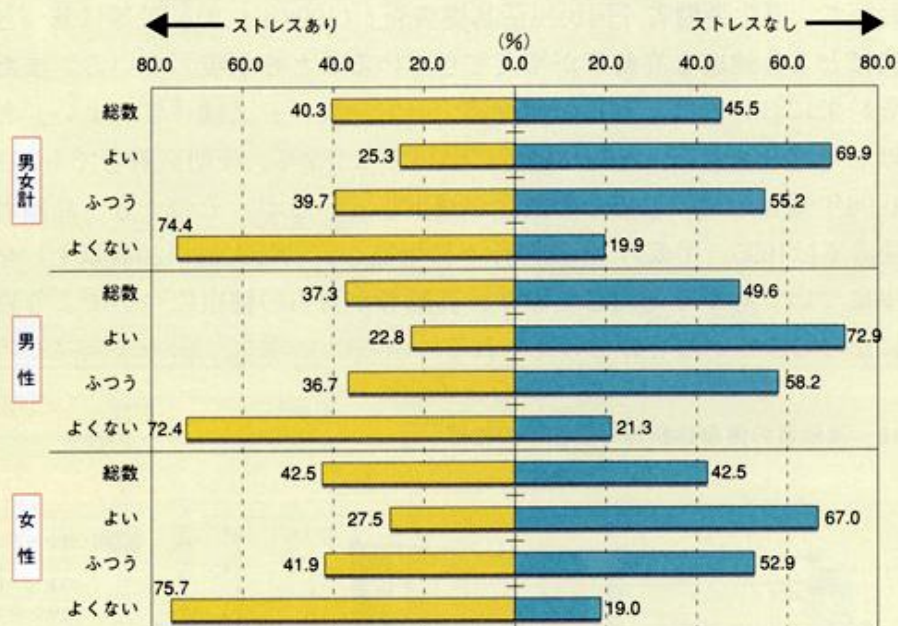


(出典) 介護保険事業状況報告 他

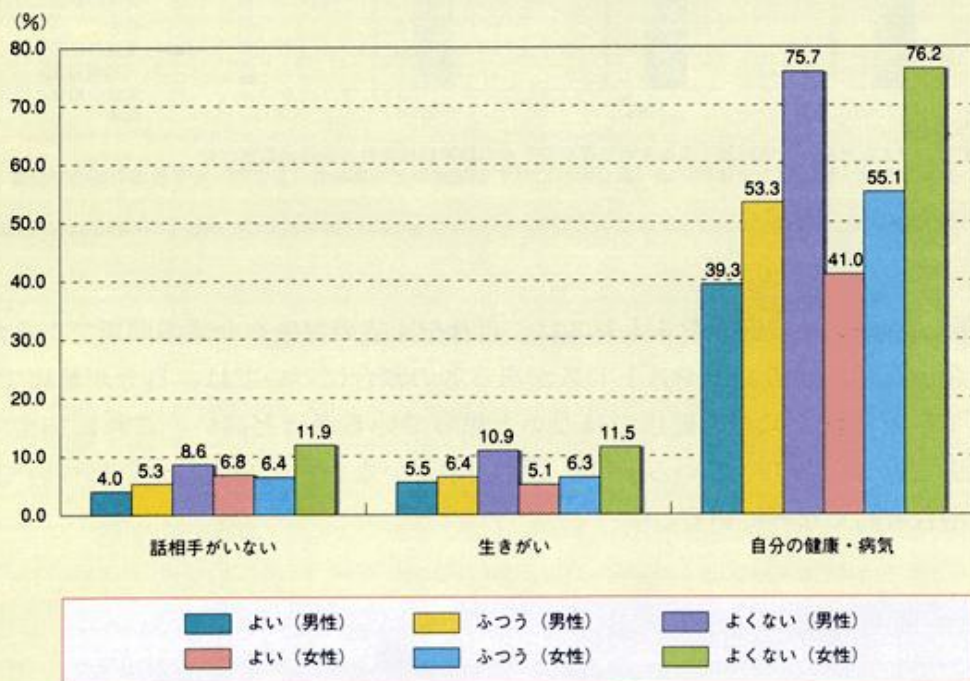
図 1.2 要介護度別認定者数の推多

従って、医療の中心は高齢者医療となりつつある。病院の年齢別利用者も高齢者が多い事は厚生労働省統計データにより明らかである。高齢になると身体に疾患を負う事も多く、それに伴い利用率も増加する。高齢者の病院受診理由は以前からの疾患以外にストレスや健康状態の維持が多い(図 1.3)。厚生労働省白書より。

図表1-1-19 高齢者の健康と悩み・ストレスの有無との関係



悩み・ストレスの種類（複数回答）



よい＝健康意識が「よい」「まあよい」の合計
 よくない＝健康意識が「あまりよくない」「よくない」の合計

資料：厚生労働省大臣官房統計情報部「国民生活基礎調査」（2001年）から政策統括官付政策評価官室作成

図 1.3 高齢者の健康と悩み・ストレスの有無との関係

自ら動ける高齢者は病院へ行けるが、介助が必要な高齢者は容易に病院へは行けない。近年では、訪問サービスが普及し始め、介助者無しでは動けない高齢者も医療福祉サービスが受けられるようになってきている。訪問サービスは主に在宅ケアサービス（専門の教育を受けたホームヘルパー・介護福祉士が直接利用者の家を訪問し、入浴・排泄・食事の介助・家事援助や通院等乗降介助といった生活全般の介護サービス）を指し、すべての65歳以上の高齢者が利用することが可能である。老人福祉施設（老人福祉法第5条3の定めより、老人デイサービスセンター・老人短期入所施設・養護老人ホーム・特別養護老人ホーム・軽費老人ホーム・老人福祉センター・老人介護支援センター）のうち、養護老人ホームに入るためには市町村に申請し、行政措置により入所が決定する。特別養護老人センター（介護老人福祉施設）は、入居を希望する本人が申請をして入居を決定する。施設により判定基準は異なり、入居者数の多さから待たされる高齢者も数多く居る。在宅ケアサービスのほかに介護支援として訪問看護、訪問医療を65歳以上の高齢者は受けられる。

1.2 問題点

在宅ケアサービスだけではカバーできない症状を抱える高齢者が存在する。病院に入るためには医療区分（厚生労働省：社会保障全般より、表1 医療区分 引用）を判定しなければならない。判定はメディカルソーシャルワーカー（社会福祉士）が行う。医療区分2・3の判定の対象となる高齢者には更に「介護（生活支援）型サービス」が主に必要としているのか、「医療的なサービス」を主に必要なのかを振り分ける。医療的なサービスの中でも、疾患の重度により高齢者を「医療群」と「非医療群」（医療群とは、点滴や投薬が医師の管理下により常時必要な状態。非医療群とは、医療群に各当しない疾患。）にグループ分けする。医療群に入らない限り、即入院とはならない。各都道府県には高齢者相談センター（シルバー110番）や在宅老人介護支援センターなど相談に応じてくれる機関が存在するが、現状では認知度が低い。家族が同居している場合、一緒に相談できるが、老老介護のように、同居している家族も高齢だった場合、入院制度や多くの保険制度の説明を、判断力や理解力の低下が進んできている高齢者に理解を求めるのは難しい。

表1 医療区分

医療区分3	<p>【疾患・状態】</p> <ul style="list-style-type: none"> ・スモン・医師及び看護師による24時間体制での監視管理を要する状態 <p>【医療処置】</p> <ul style="list-style-type: none"> ・中心静脈栄養 ・24時間持続点滴 ・レスピレーター使用 ・ドレーン法・胸腹腔洗浄 ・発熱を伴う場合の気管切開、気管内挿管のケア ・酸素療法 ・感染隔離室におけるケア
医療区分2	<p>【疾患・状態】</p> <ul style="list-style-type: none"> ・筋ジストロフィー ・多発性硬化症 ・筋萎縮性側索硬化症 ・パーキンソン病関連疾患 ・その他神経難病（スモンを除く） ・神経難病以外の難病 ・脊髄損傷 ・肺気腫・慢性閉塞性肺疾（COPD）・疼痛コントロールが必要な悪性腫瘍 ・肺炎 ・尿路感染症・創感染・リハビリテーションが必要な疾患が発症してから30日以内 ・脱水 ・体内出血 ・頻回の嘔吐 ・褥瘡 ・うっ血性潰瘍 ・せん妄の兆候 ・うつ状態 ・暴行が毎日みられる状態 <p>【医療処置】</p>

	<ul style="list-style-type: none"> ・透析　・発熱又は嘔吐を伴う場合の経管栄養　・喀痰吸引 ・気管切開・気管内挿管のケア　・血糖チェック　・皮膚の潰瘍のケア ・手術創のケア　・創傷処置　・足のケア
医療区分 1	医療区分 2・3 に該当しない者

2 章 医療情報化と高齢者用医療端末の現状

2.1 IT の医療化

家庭用医療情報サービスの現状を踏まえながら IT 医療について述べる。

参考文献によると、IT 医療とは「インターネットを基盤にした医療・健康・福祉情報サービスのことであり、インターネットによる病院医療の一定“質”と“良質の患者”を確保するために病院側が発足させた医療情報サービスのことである。この言葉はアメリカを中心に 1990 年後半に新しく誕生した用語である。」インターネットの持つ、普遍性、公共性、連続性、位置透過性、対話性、といった多くの特徴は、もっとも有効に利用される可能性のある分野である。しかし、一般的にインターネットの環境で大きな課題として取り扱われる情報の機密性、認証性、安全性や、既存の社会制度との整合性などの問題点が医療関係の中で、技術の分野、社会制度の分野、経済の分野など、さまざまな分野の専門家と一緒に取り組んでいかねばならない。厚生労働省（旧厚生省）における主要な情報化推進対策の中で、特に関連のあるものをあげる。（表 2.1）

表 2.1 厚生労働省（旧厚生省）における主要な情報化推進対策

1. 保健医療情報システム検討会報告（平成 6 年）
2. 厚生省情報化推進本部設置（平成 6 年）
3. G7 の世界情報通信基盤構築プロジェクトへの参加（平成 7 年）
4. 電子カルテの開発（平成 7 年度より）
5. レントゲン画像等電子保存（平成 8 年度より）
6. 国立病院情報ネットワーク（がん・循環器整備中）
7. 医療情報電子検索サービス（平成 7 年度より）
8. 健康情報ネットワーク（平成 8 年度）

このように、厚生労働省の取り組みも緩やかにだが進んでいる。

2.2 医療情報化

現在の医療はその多くが情報化している。薬の配分から個人情報、全ての医療情報管理は情報システムを利用し、データベース化し、医療行為を行う医者、看護師、薬剤師などが共通WANでつながっている。患者自身も情報システムを利用することが増えてきている。参考文献[1]によると医療情報化とは、「保険・医療サービスの効率化や迅速化、手続きの簡素化、利用者の利便を図るべく、遠隔医療用のテレビ電話、カルテ（診療録）などの医療情報の開示を、紙媒体を削減し、電子化したものである。」とある。この医療情報化に伴い、電子カルテシステムが登場した。参考文献[2]による電子カルテシステムとは「診療記録等の診療情報を電子化し、保存・更新するシステムである。」とある。参考文献[2]によると電子カルテシステムは、「医師が診療時に患者のプロファイルをすばやく把握するため、統括的な診療録に病歴や診療方針、行動制限・生活指示オーダー、不穏時・不眠時処方などが記録され、PCブラウザを使い、容易に閲覧できるようになっている。閲覧画面は、シンプルな画面デザインで工夫され、ボタンやメニューを極力軽減し、業務の効率化とスピードアップが図られている。また、このカルテ情報を使い、医師、看護師、コメディカルなど異なる職種間で患者情報を共有しているのである。」とある。

第3章 家庭における医療情報化

この章では既存の家庭医療情報化、情報端末の現状を述べる。

3.1 家庭医療端末の現状

家庭医療端末・在宅医療システムとして多く利用されているのは遠隔診療支援システム（遠隔医療用テレビ電話）である。

遠隔診療支援システムとは都市部の医療機関が、医療過疎状態にある遠隔地の医療機関とネットワークで結ばれ、専門的な加増診断、病理診断などの医療支援を行うシステムである。参考文献[5]によると、「福島県葛尾村では平成11年4月から、村内にある葛尾村診療所と葛尾村の高齢者が多く通う村外の8つの病院でテレビ電話診療を実施している。（参考文献より、図3.1 引用 参照）

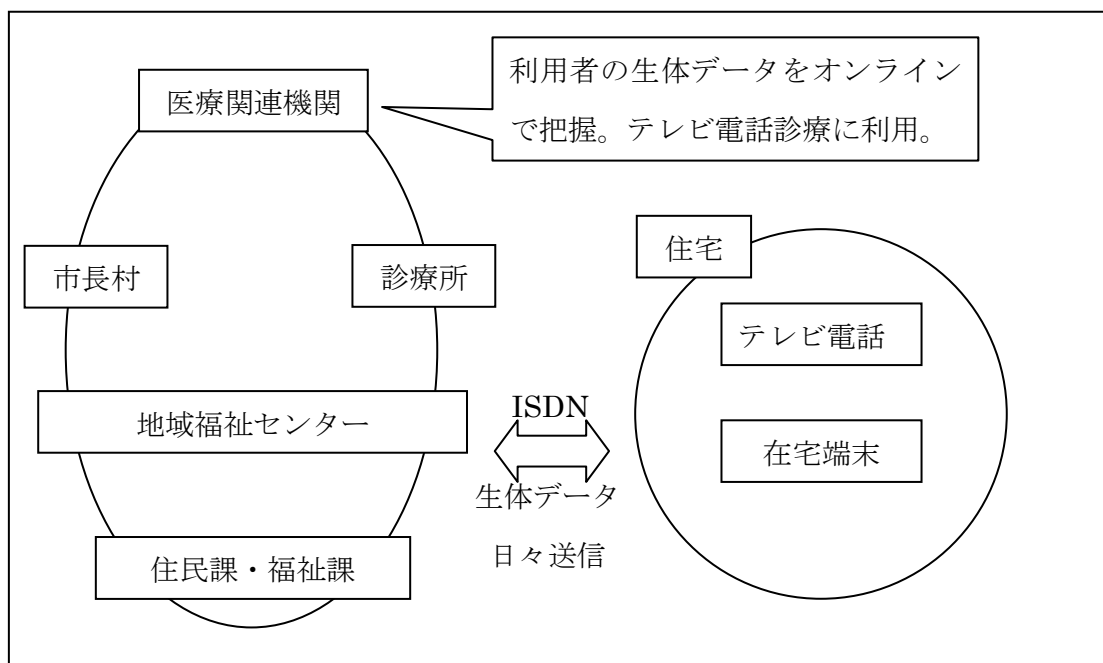


図 3.1 遠隔診療支援システム

具体的には、テレビ電話による遠隔診療と薬の宅配が行われている。これにより患者側は通院回数を大幅に減らすことができ、通院に要する体力的、経済的、精神的負担が軽減された。医師側においても、①スケジュールに基づいて定期的に診療を行うことが出来るため、慢性患者を治療する医療側としては、常に管理下にあるという安心感がある。そして②診療中断による不測の事故(高血圧患者の脳血管障害など)を予防できるようになる、③テレビ電話の画面をみながら直接本人に話しかけることができるため、病状の変化を感

知することが容易で、来院が必要な状態なのかどうか、客観的に把握することが可能となったなどのメリットがあげられている。」とある。このように日々の健康データを保健師が確認し、健康を損ねるような数値がみられた場合など、早期に利用者宅を訪問し、健康相談や栄養指導など、疾病の予防や早期発見などの健康管理にも役立つと考えられる。これは、自宅だけではなく、外にでても利用できる非常に便利で簡易な技術である。しかし、便利で簡易と思い利用しているのは情報化技術に多少の知識や理解をしめしている者たちだけで、デジタルディバイト（パソコンやインターネットなどの IT(情報技術)を使いこなせる者と使いこなせない者の間に生じる、待遇や貧富、機会の格差。コンピュータを使いこなせない高齢者や貧困のため情報機器を入手できない人々は、より一層困難な状況に追い込まれ、情報技術が社会的な格差を拡大、固定化する現象がデジタルディバイドである。）が存在している。医療情報化に伴い、情報インフラ（何らかのシステムや事業を有効に機能させるために基盤として必要となる設備や制度などのこと。）の改善に重点を置き、高齢者も IT を生活に導入できる事が必要である。

3.2 セルフメディケーション

セルフメディケーションとは、自分自身で健康を管理し、軽い病気の症状の緩和や予防のために、自分の責任で医薬品などを使って手当をする事である。風邪を引いたとき、薬局やドラッグストアなどで求めた総合かぜ薬を飲んで治す、これがセルフメディケーションである。自分の健康は自分で守るという意識をもって、毎日の生活を快適に過ごす、これこそがセルフメディケーションの実践といえる。なぜ今セルフメディケーションが必要なのか、それは、自分の健康を他人任せにできない、医師にすべてを委ねる医療ではなく、患者自らの治癒への意思と努力が求められているからである。人生の中で病気や障害の無い期間、つまり健康寿命をいかに延ばすかが課題なのである。自分の健康を自分で守り、維持・増進させるためには、セルフメディケーションが必要となる。家庭医療端末が普及し始めている理由は、自分自身でも健康管理をする必要があるからである。高齢者にはセルフメディケーションが必要なのは、1章で提示した「病院受診理由」の中の精神的ストレスの部分である。自分の病気を家族や近い人に相談するのではなく、医師に相談している。自分の力で健康を守る力が弱くなってきている。自分の病気について、医師に相談する事は悪いことではない。しかし、自分自身で出来る健康維持の方法を見つけなければ、不安は改善できない。血圧や体温など、自宅でも健康状態を把握できる医療機器は多く販売されている。すべてを医師・医療機関に頼らずとも自分の健康状態を調べる事は可能である。

4 章 家庭医療端末の情報化

既存のシステム、家庭用医療端末は市販されていない。家族が居る家庭向け医療システムは、徘徊見守りサービスや、ドア開閉感知システムが開発されているが、家庭用で家族や高齢者自身が触って使う端末はない。また、高齢者の能力に合ったシステムになっていない。在宅医向け（往診システム 厚生労働省報告 情報システム、在宅医向けシステム）も個人向けとして開発はされていない。家族や医師が使うことを想定されて開発されているため、高齢者自身が使用しない。それでは老老介護や独居老人（子や孫などの親族と居を別にし、配偶者もいない状態で暮らしている高齢者のこと）はシステムを利用できない。家族と暮らしている高齢者も多いが、独居老人数も年々増加している。（図 4.1 厚生労働省）

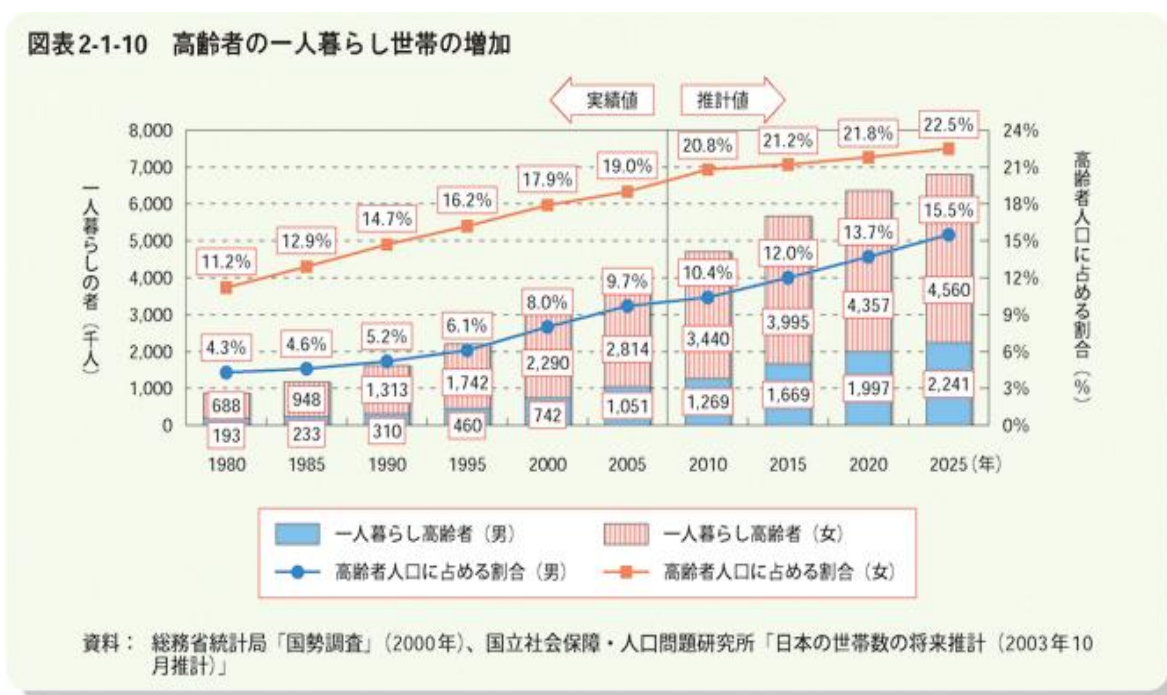


図 4.1 高齢者の独り暮らし世帯の増加

個人宅に家庭用医療端末は無いが、医療機器である、血圧計と体温計、心電図計測器を持っている高齢者や一般家庭は多い。しかし、毎日結果を取らせるだけでは、大きな変化は無いにせよ、いずれ効果がでないと飽きてしまう。遠隔地では病院に行く回数も多くはない。身体記録の結果を持て余してしまう。セルフメディケーションの重要性は、自己管理である。結果を持て余してしまうのは、コメントをしてくれる医師や家族が傍に居ない事

が原因である。汎用システムをもっと家庭用に安く、代用する必要がある。IT ディバイドでも、誰でも使いやすいインタフェースが求められている。市町村での導入が未だ少なく、家庭での簡易医療に重点を置くべきである。セルフメディケーションと情報化の複合が必要である。

5 章 解決へのアプローチ

5.1 専用から汎用へ

専用機器は家庭向けではない、それならば、専用機器を汎用品を使い家庭用にする。

そのためには、専用機のコストパフォーマンスを抑え、高齢者や家族が使いやすい、簡易なシステム化が鍵となる。高齢者に使えないから IT ディバイドになるのであって、使いやすく容易であれば、ディバイドとはならない。高齢者が使いやすいものとは、表示する字が大きく、音声で説明をし、操作が単純なことが必要である。使用者の高齢者が個人情報端末を利用し、既存の体温計や血圧計を使い入力した身体データを主治医と離れた家族に送信する。主治医はそのデータをみて個人別の対応をする。

個人端末には ID を振り分け、本人を認証するシステム（どんな？）とすれば個人情報を保護しながら、生体データの取り扱いが可能となる。

6 章 設計と実装

本章では、前章までの議論を踏まえ、情報技術を活用することで高齢者が自宅にいながらにして、遠隔の病院にいる医者から日常的に健康状態を管理できる基礎データ収集の仕組みを検討する。仕組みは、汎用的で低コストであるとともに、デジタルディバイドである高齢者でも容易に扱えることを目指して、プロトタイプシステムの設計実装を行った。

6.1 設計

6.1.1 設計目標

前章までの議論から、特に高齢者を対象とした医療における多くの問題点が指摘された。日常の家庭で自ら健康管理を行い、その情報収集と医療スタッフによるアドバイスや管理体制を実現するというセルフメディケーションへの流れは、医療情報化の傾向の中で、特に情報技術を活用することで実現がより行われやすいと考えられる。しかし、急速な高齢化社会でセルフメディケーションが必要となるのは、高齢者層が中心となるのが自明であるにも関わらず、既存の専用システムが価格や運用コストが高く、インタフェースの面では操作が困難であるなど、介護してくれる家族がおらず独居も多い社会的弱者であるとともにデジタルディバイトである高齢者が容易に導入、利用できる形にはなっていないことが明らかになった。

そこで、本研究では、このような問題点に対して、以下のような設計目標を設定することとした。

- 高齢者の専用のハードウェアや専用回線ではなく、安価な市販の製品や通信回線を利用できること。
- ソフトウェアも基盤となる部分について、Web 等の汎用的なインタフェースを利用しつつ、画面構成などの工夫により、高齢者が本人一人でも操作に臆することなく、直感的に入力できること、また日常的に継続できること。
- 高齢者本人のデータを本人だけでなく、遠隔の介護・医療関係者や親族などの関係者が容易に閲覧できること。
- データの閲覧では、必要に応じて現在、過去の履歴を視覚化して表示するなどの機能を設けること。また、異常値など、注視すべき変化があった場合に、通知できること。

6.1.2 構成要素と機能

このような設計目標と、前章までの議論および高齢者のためのセルフメディケーションモデルから、プロトタイプシステムの構成要素と機能について検討した。

- 構成要素・登場人物

①介護される人（高齢者本人）：情報機器の操作の援助を行える家族などの要員が必ずしもいない家庭を想定する。

②介護・医療関係者：①の健康状態について管理し指導を行う立場の者。①がよく通う、またはかかりつけの医者や看護師、介護ヘルパーなどの関係者を指す。

③家族などの親族：直接介護できる、できないに関わりなく、①の状態を気にかけて、必要な場合に駆けつけることができるなど、①の健康、生命の状態について、それなりに責任がある者を指す。

- 健康管理のための個人データの種類

- 体温、血圧、元気レベル

本研究の実装では、最低限必要と思われる健康管理データで、かつ既存（手持ち）の装置によるデータ取得が可能なものを選択した。技術の進化による自動化も将来的には期待される。送信すべき個人データの種類が増えた場合には、自動化が必要となると考えられる。

- 機能

- 個人特定機能

本システムを利用する高齢者をID等で識別するとともに、パスワード等で認証することによって、なりすまし等の脅威を回避し、個人の健康管理データの流出、漏洩などを防止する。高齢者が利用しやすい本人確認機能については、多くの議論があり、本章の設計目標では優先度が高くないため、今回の設計実装ではこのような仕組みがあること前提として検討を進める。プロトタイプシステムには実装しない。

- 体温入力機能

体温計で測ったデータを入力する機能である。近い将来可能であれば、コンピュータに接続可能なデジタル入出力機能を持つ体温計が登場する可能性もあるが、高齢者にとってはデジタルというだけで利用する以前に壁を感じてしまうこともある。また、長年使い続けてきた既存の体温計になじんでおり、十分に使いこな

すことができる。要は、体温を測ってもらい、それを入力してもらうことでメリットにつながることを実感してもらうことも目標の一つとなる。したがって、本機能においては、直感的に入力してもらえるインタフェースを備えることと、入力されたデータが収集するサーバに正しく送信されることが要求される。

➤ 血圧入力機能

血圧計で測ったデータを入力する機能である。体温計と同様であるが、体温計と異なり血圧計については、コンピュータに直接接続できないまでも、デジタル化された機器が、多くの高齢者宅に普及しているそうである。したがって、本機能でも、使い慣れた血圧計を利用してもらい、計測された血圧を視認し、入力してもらうことが重要である。また、同様に入力されたデータが収集するサーバに正しく送信されることは当然である。血圧には、最高値と最低値を計測することとなっており、この2つのデータを利用する。

➤ 元気レベル入力機能

本人の自覚症状を確認する健康状態を元気レベルとして入力する機能である。これは、体温や血圧と異なり、本人がどのように感じているかを入力してもらうこととなる。数値で判断できるデータだけでなく、本人の感覚というアナログなデータからしか捉えられない事象や症状もあると考えられ、5段階で入力してもらうこととする。レベルとその内容を表 6.1 に示す。レベルは0を最高とし、1から加点式で健康マイナスの評価とする。

表 6.1：元気レベルの内容

レベル	内容
0	元気
1	まあまあ
2	いつも通り
3	少し悪い
4	体調不良

➤ 入力データ送信機能

上記の機能で入力された健康管理データを一括して送信する機能である。本プロトタイプシステムでは、体温、血圧、および元気レベルは、例えば、朝起床時などにすべて計測して申告してもらうことを想定している。健康管理するためのデータであるので、取得する時間にばらつきがないことが望ましいと考えられるためである。

➤ データベース登録機能

利用者から入力されサーバに送信されたデータを受信し、データベースに登録する機能である。この際、利用者を識別する情報、健康管理データ（体温、血圧、元気レベル）、登録時刻を組にして登録する。

➤ 個人データ管理機能（医療・介護関係者用）

医療・介護関係者においてそれぞれの利用者の担当が、登録されたデータを閲覧するだけでなく、必要に応じて、日々の変化や経過を観察し、留意点等を追加で入力し、カルテのように個人のデータを管理する機能を持つ。例えば、管理機能としては、以下のようなものが考えられる。

◇ 最新情報一覧

医者や医療スタッフが担当する利用者は一人ではない場合も多いと考えられるため、最新の登録データを一度に閲覧できることが期待される。

◇ 履歴表示

1週間、1ヶ月、3ヶ月、6ヶ月、1年といった期間で、各個人の履歴を閲覧でき、個々人の健康状態の分析管理に役立てることが可能である。また、視覚化の観点では、各人の履歴をグラフ等で表示するなどの、既存の多くの方法を比較して導入することを検討する。

◇ 警告

あらかじめ設定された各人の危険数値から大きく離れたデータが登録されたときに、動的にメール等で通知する。

➤ 特定個人データ閲覧機能（家族等の親族用）

介護される高齢者本人の健康状態に責任のある家族等の親族が、登録されたデータを閲覧する機能である。介護・医療関係者が入力、管理する病状や経過観察データなどのカルテデータは閲覧させず、予め登録したデータだけを許可された者

のみが閲覧できるようにする。

6.1.3 システム構成

モデルをベースに、本章で検討した各機能がどのように配置され、どの構成要素によって、どのような流れで処理されるのかを図 6.1.3.1 に示す。

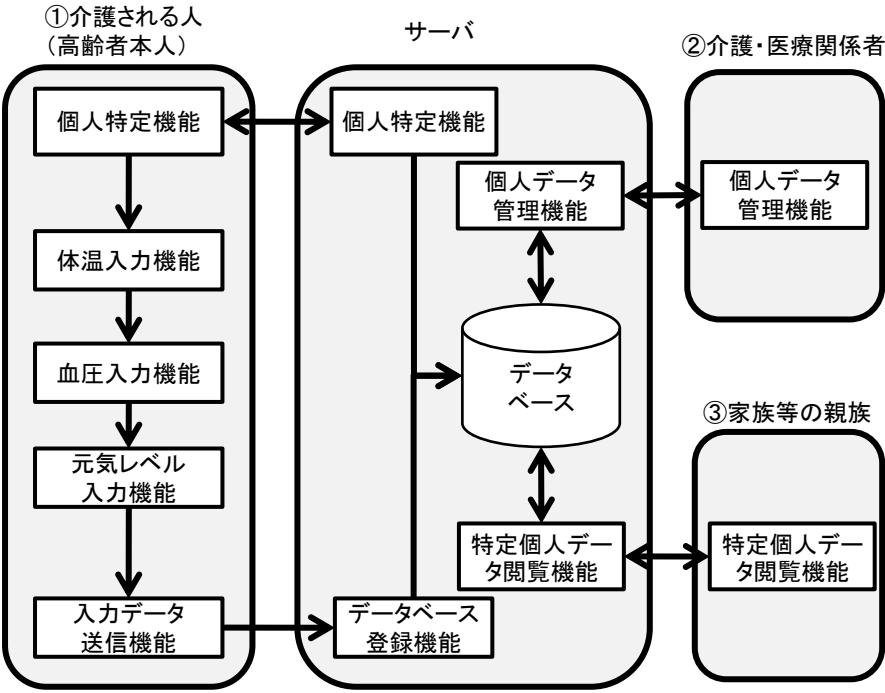


図 6.1.3.1 システム構成

6.1.4 テーブルの設計

本プロトタイプシステムにおいて、各機能によって利用され、データベースに登録される情報の種類と内容に基づいて、テーブルを設計する。表 6.1.4.1 にテーブルの項目内容、フィールド名、属性を示す。

表 6.1.4.1 テーブルの設計

項目内容	フィールド名	属性
個人 ID	Id	char(30)
入力時刻	Jikoku	Datetime
血圧最高値	Ketuatumax	int(20)
血圧最低値	Ketuatumin	int(20)
体温	Taion	float
元気レベル	Bodylevel	int(11)
連番	Renban	int(11)

このテーブルによってデータベースに登録されるレコードの重複を防止するために、レコードに対して唯一となる通し番号を割り当てることとする。このフィールドを連番として定義する。

6.1.4 入力画面インタフェースの設計

設計目標でも述べたように本研究では「Web 等の汎用的なインタフェースを利用しつつ、画面構成などの工夫により、高齢者が本人一人でも操作に臆することなく、直感的に入力できること、また日常的に継続できること」を目標とする。特に、高齢者に操作を求める各種データの入力機能においてこの目標達成が要求される。

Web を利用した場合の入力画面としては、例として、フォームがある。フォームでは、HTML のタグで記述される入力インタフェースであり、テキストやパスワードの入力欄、ラジオボタン、チェックボックスなどが利用できる。例を図 6.1.4.1 に示す。実現方法がシンプルであるので利用しやすいが、高齢者にとってやさしいか、入力しやすいかは試して見る必要がある。本研究では、比較的操作の面で負担が少なそうなラジオボタンを利用した入力インタフェースを体温入力画面で試作することとする。

図 6.1.4.1 フォームによる入力インタフェースの例

同様に、画像などで作成したボタンにリンクを張ることで、画面上に構成されたテンキーをクリックして、入力してもらう方法もある。

その他にも、JavaScript や Flash を利用して、同様なことを実現する技術もあるが、どの技術も技術それ自体が大事なのではなく高齢者にとって使いやすいかどうかが鍵となる。

設計するにあたって、高齢者の日常生活の中ですでに定着あるいは利用しやすいとされる入力インタフェースを調べたところ、銀行や郵便局などの ATM が代表としてあげられた。特に、過疎化が進む農村部や地方都市では、ゆうちょ銀行（旧郵政グループの郵便貯金）の ATM が多く使用されている。振り込み画面の例を図 6.1.4.2 に示す。



図 6.1.4.2 ゆうちょ銀行の ATM 画面（ゆうちょ銀行ホームページより）

このなじみのあるインタフェースを模して作成することにより、高齢者が躊躇せずに直感的に入力できる可能性があると考え、血圧、元気レベルについては、このテンキーインタフェースを参考にして、画面に対して大きめのボタンを使用して、入力画面を作成することとする。

6.2 実装

前節までのプロトタイプシステムの設計に基づき、実装を行う。

6.2.1 実装環境

本研究では、プロトタイプシステムの実装を以下のような実装環境で行う。

● ハードウェア

<ul style="list-style-type: none">・サーバ IBM ThinkPad X31 CPU : Pentium M 1.6GHz メモリ : 1GB HDD : 60GB	<ul style="list-style-type: none">・クライアント Windows PC CPU: Pentium(R) M 1.7GHz メモリ: 1.5GB HDD: 80G
--	--

● ソフトウェア

<ul style="list-style-type: none">・サーバ オペレーティングシステム : Ubuntu 8.04 Server Web サーバ : Apache 2.2.8 データベース : MySQL5.0.51 プログラミング言語 : PHP5.2.4	<ul style="list-style-type: none">・クライアント オペレーティングシステム : Microsoft Windows XP Home Edition Vesion2002 Service Pack 3 Web ブラウザ : Mozilla FireFox, Internet Explorer
---	--

6.2.2 データベースサーバ MySQL における設定

● データベースの作成

データベースサーバ MySQL において、MySQL モニタによりデータベース sotu_a640011 の作成を行った。この操作と結果を図 6.2.2.1 に示す。

```
mysql> create database sotu_a640011;  
Query OK, 1 row affected (0.00 sec)
```

図 6・2・2・1 データベースの作成

● テーブル作成およびフィールドの設定

同様に、設計に基づいたフィールドの設定を含む、テーブル `kojindata` の作成を行った。この操作と結果を図 6.2.2.2 に示す。

```
mysql> use sotu_a640011;
Database changed
mysql> create table kojindata ( renban int, id char(30) , jikoku datetime ,
ketuatumax int(20) , ketuatumin int(20) , taion float , bodylevel int(11) , renban
int(11));
Query OK, 0 rows affected (0.00 sec)
```

図 6.2.2.2 テーブルの作成

- 連番設定

また、重複防止用に作成した連番フィールドに対して、インデックス化と自動加算の設定を行った。この操作と結果を図 6.2.2.3 に示す。

```
mysql> alter table kojindata add index ind1(renban);
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> alter table kojindata modify renban int auto_increment;
Query OK, 0 rows affected (0.00 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

図 6.2.2.4 連番設定

6.2.3 機能の実装

本研究では設計で述べた機能のうち、高齢者が躊躇せずに直感的に入力できるようなインタフェースを実現することが目標となっているため、健康管理データを入力、送信、登録するための以下の機能を実装することとした。

- 実装機能
 - 体温入力機能
 - 血圧入力機能
 - 元気レベル入力機能
 - 入力データ送信機能
 - データベース登録機能

これらの機能を、高齢者側では、クライアントとして汎用的な Web の入力インタフェースを構築するとともに、POST や GET などの http の送信機能を主に使用してデータをサーバに送信し、サーバ側では受信したデータをデータベースに登録するものとした。

- プログラムの構成

図 6.2.3.1 に以上の機能の構成を示す。

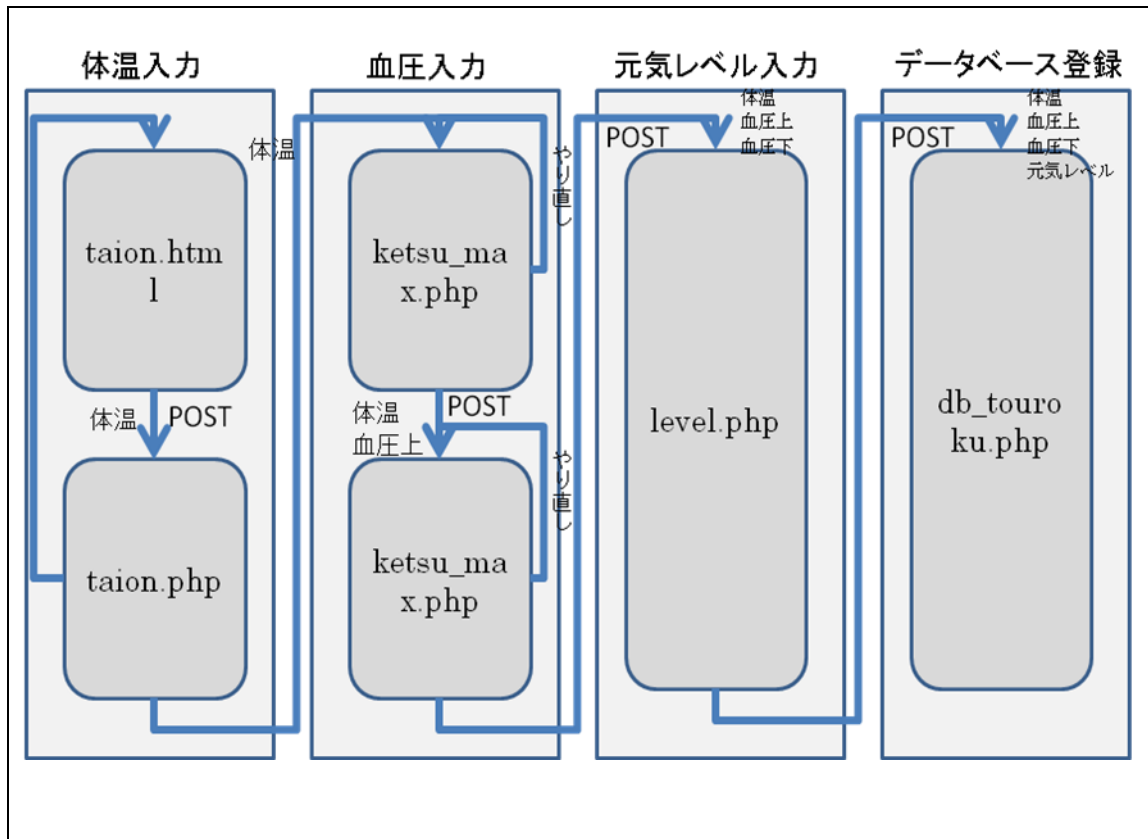


図 6.2.3.1 プログラムの構成

- 体温入力機能

taion.html および taion.php で構成される。

taion.html ではフォームにより、ラジオボタンで体温の上二桁と下一桁を分けて選択し、それらを POST で taion.php に送信する。

taion.php では、受信した体温を表示し、確認後、血圧入力機能の ketsu_max.php に POST で送信する。

それぞれのプログラムのソースを図 6.2.3.2、図 6.2.3.3 に示す。

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
  <head>
    <title>体温データ入力</title>

```

```
<meta http-equiv="Content-Type"
      content="text/html; charset=Shift_JIS">

</head>

<body>

<form method="post" action="taion.php">

<p>体温は何度台ですか？ <br>
<input type="radio" name="q1" value="35"> 35 度台<br>
<input type="radio" name="q1" value="36"> 36 度台<br>
<input type="radio" name="q1" value="37"> 37 度台<br>
<input type="radio" name="q1" value="38"> 38 度台<br>
<input type="radio" name="q1" value="39"> 39 度台<br>
<input type="radio" name="q1" value="40"> 40 度台<br>
<input type="radio" name="q1" value="41"> 41 度台<br>
<input type="radio" name="q1" value="42"> 42 度台<br>
</p>

<p>何分？ <br>
<input type="radio" name="q2" value="0.0"> 0 分<br>
<input type="radio" name="q2" value="0.1"> 1 分<br>
<input type="radio" name="q2" value="0.2"> 2 分<br>
<input type="radio" name="q2" value="0.3"> 3 分<br>
<input type="radio" name="q2" value="0.4"> 4 分<br>
<input type="radio" name="q2" value="0.5"> 5 分<br>
<input type="radio" name="q2" value="0.6"> 6 分<br>
<input type="radio" name="q2" value="0.7"> 7 分<br>
<input type="radio" name="q2" value="0.8"> 8 分<br>
<input type="radio" name="q2" value="0.9"> 9 分<br>
```

```

</p>

<p><input type="submit" value="確認"></p>

</form>

    </body>

</html>

```

図 6.2.3.2 taion.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
    <head>
        <title>体温確認・登録</title>
        <meta http-equiv="Content-Type" content="text/html;
charset=Shift_JIS">
    </head>
    <body>
<?php
extract($_POST);
extract($_GET);

//レコード
if ($taion<35 && $taion >= 43){
    echo "体温データが正しく入力されていません";
    echo "<p><a href=taion.html>体温データ入力に戻る</a></p>";
    exit;
} else {

//確認のためのフォーム

```



```

        echo "<form action = ¥\"ketsu_max.php¥\" method=¥\"post¥\">";
        echo "<p>あなたの体温は";
        $taion = $q1+$q2;
        echo $taion;
        echo "ですね? </p>";

        echo "<input type = ¥\"hidden¥\" name = ¥\"taion¥\" value =¥\"\"";
        echo $taion;
        echo "¥\">";

        echo  "<p><input    type=¥\"submit¥\"    name=¥\"kakunin¥\"
value=¥\"血圧入力へ¥\">";
        echo "<p><a href=¥\"taion.html¥\">戻る</a>";
        echo "</form>";

    }

?>
</body>
</html>

```

図 6.2.3.3. taion.php

- 血圧入力機能

ketsu_max.php および ketsu_min.php で構成される。

ketsu_max.php では、ゆうちょ銀行の A T M画面を模した数字アイコンで構成されたテンキーを表示し、それぞれを繰り返しクリックすることで、自分自身に GET で送信し、血圧データを入力する。確定後 POST で ketsu_min.php に送信する。

ketsu_min.php では、ketsu_max.php と同様の動作を行い、値が確定後、元気レベル入力機能の level.php に POST で送信する。ここまで、体温データも合わせて送信される。それぞれのプログラムのソースを図 6.2.3.4、図 6.2.3.5 に示す。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
    <head>
        <title>血圧(上)入力</title>
        <meta http-equiv="Content-Type"
            content="text/html; charset=Shift_JIS">
    </head>
    <body>
<?php
extract($_POST);
extract($_GET);

$ketsu_max .= $input;

echo "体温: $taion <br>";
echo "
<p>次に、あなたの血圧の上はいくつですか？数字を押してください。</p>
<form action=¥"ketsu_min.php¥"  method=¥"post¥" >
<p> 血 圧    ( 上 )    :    <input    type=¥"text¥"    name=¥"ketsu_max¥"
value=¥"$ketsu_max¥" size=¥"40¥"></p>
```

```

<input type = ¥"hidden¥" name = ¥"taion¥" value =¥"$taion¥">
<p><input type=¥"submit¥" value=¥"終了¥">
<a href=¥"ketsu_max.php?taion=$taion¥">やり直し</a></p>
</form>

<table border=¥"1¥" align=¥"center¥" width=¥"300¥" height=¥"130¥">
<tr align=¥"center¥">
<td><a
href=¥"ketsu_max.php?input=1&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/01.jpg¥"></a></td>
<td><a
href=¥"ketsu_max.php?input=2&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/02.jpg¥"></a></td>
<td><a
href=¥"ketsu_max.php?input=3&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/03.jpg¥"></a></td>
</tr>
<tr align=¥"center¥">
<td><a
href=¥"ketsu_max.php?input=4&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/04.jpg¥"></a></td>
<td><a
href=¥"ketsu_max.php?input=5&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/05.jpg¥"></a></td>
<td><a
href=¥"ketsu_max.php?input=6&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/06.jpg¥"></a></td>
</tr>
<tr align=¥"center¥">

```

```

<td><a
href=¥"ketsu_max.php?input=7&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/07.jpg¥"></a></td>

<td><a
href=¥"ketsu_max.php?input=8&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/08.jpg¥"></a></td>

<td><a
href=¥"ketsu_max.php?input=9&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/09.jpg¥"></a></td>
</tr>
<tr align=¥"center¥">
<td><a
href=¥"ketsu_max.php?input=0&ketsu_max=$ketsu_max&taion=$taion¥"><img
src=¥"img/00.jpg¥"></a></td>
</tr>
</table>
</body>
</html>
";

?>

```

図 6.2.3.4 : ketsu_max.php

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
  <head>
    <title>血圧(下)入力</title>
    <meta http-equiv="Content-Type"
content="text/html; charset=Shift_JIS">

```

```

        </head>

        <body>

<?php
extract($_POST);
extract($_GET);

$ketsu_min .= $input;

echo "体温: $taion  ";
echo "血圧(上): $ketsu_max <br>";

echo "
<p>次に、あなたの血圧の下はいくつですか？数字を押してください。</p>
<form action=¥"level.php¥"  method=¥"post¥" >
<p>血圧（下）: <input type=¥"text¥" name=¥"ketsu_min¥" value=¥"$ketsu_min¥"
size=¥"40¥"></p>
<input type = ¥"hidden¥" name = ¥"taion¥" value =¥"$taion¥">
<input type = ¥"hidden¥" name = ¥"ketsu_max¥" value =¥"$ketsu_max¥">
<p><input type=¥"submit¥" value=¥"終了¥">
<a href=¥"ketsu_max.php?taion=$taion&ketsu_max=$ketsu_max¥"> や り 直 し
</a></p>
</form>

<table border=¥"1¥" align=¥"center¥" width=¥"300¥" height=¥"130¥">
<tr align=¥"center¥">
<td><a
href=¥"ketsu_min.php?input=1&ketsu_min=$ketsu_min&ketsu_max=$ketsu_ma
x&taion=$taion¥"><img src=¥"img/01.jpg¥"></a></td>
<td><a

```

```

href=¥"ketsu_min.php?input=2&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/02.jpg¥"></a></td>

<td><a
href=¥"ketsu_min.php?input=3&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/03.jpg¥"></a></td>

</tr>

<tr align=¥"center¥">

<td><a
href=¥"ketsu_min.php?input=4&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/04.jpg¥"></a></td>

<td><a
href=¥"ketsu_min.php?input=5&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/05.jpg¥"></a></td>

<td><a
href=¥"ketsu_min.php?input=6&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/06.jpg¥"></a></td>

</tr>

<tr align=¥"center¥">

<td><a
href=¥"ketsu_min.php?input=7&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/07.jpg¥"></a></td>

<td><a
href=¥"ketsu_min.php?input=8&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/08.jpg¥"></a></td>

<td><a
href=¥"ketsu_min.php?input=9&ketsu_min=$ketsu_min&ketsu_max=$ketsu_max&taion=$taion¥"><img src=¥"img/09.jpg¥"></a></td>

</tr>

<tr align=¥"center¥">

```

```

<td><a
href=¥"ketsu_min.php?input=0&ketsu_min=$ketsu_min&ketsu_max=$ketsu_ma
x&taion=$taion¥"><img src=¥"img/00.jpg¥"></a></td>
</tr>
</table>
</body>
</html>
";
?>

```

図 6.2.3.5 : ketsu_max.php

- 元気レベル入力機能・入力データ送信機能

level.php で構成される。フォームのラジオボタンで元気レベルを選択し、体温、血圧と合わせて、サーバのデータベース登録機能の db_touroku.php に送信される。このプログラムのソースを図 6.2.3.6 に示す。

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
  <head>
    <title>元気レベル</title>
    <meta http-equiv="Content-Type"
    content="text/html; charset=Shift_JIS">
  </head>
  <body>
<?php
extract($_POST);

```

```

echo "体温: $taion ";
echo "血压(上): $ketsu_max ";
echo "血压(下): $ketsu_min <br>";

echo "
<form method=¥"post¥" action=¥"db_touroku.php¥">
<p>次に、あなたの元気レベルはどれですか？</p>

<input type=¥"radio¥" name=¥"level¥" value=¥"0¥"> 元気<br>
<input type=¥"radio¥" name=¥"level¥" value=¥"1¥"> まあまあ<br>
<input type=¥"radio¥" name=¥"level¥" value=¥"2¥"> いつも通り<br>
<input type=¥"radio¥" name=¥"level¥" value=¥"3¥"> 少し悪い<br>
<input type=¥"radio¥" name=¥"level¥" value=¥"4¥"> 体調不良<br>

<input type = ¥"hidden¥" name = ¥"taion¥" value =¥"$taion¥">
<input type = ¥"hidden¥" name = ¥"ketsu_max¥" value =¥"$ketsu_max¥">
<input type = ¥"hidden¥" name = ¥"ketsu_min¥" value =¥"$ketsu_min¥">

<p><input type=¥"submit¥" value=¥"終了¥">
</p>
</form>

</body>
</html>
";

?>

```

図 6.2.3.6 level.php

- データベース登録機能

db_touroku.php で構成される。受信した体温、血圧、元気レベルの各データを ID と取得時刻など合わせて、データベースに登録する。このプログラムのソースを db_touroku.php を図 6.2.3.7 に示す。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html lang="ja">
    <head>
        <title>元気レベル</title>
        <meta http-equiv="Content-Type"
            content="text/html; charset=Shift_JIS">
    </head>
    <body>
<?php
extract($_POST);

echo "体温: $taion  ";
echo "血圧(上): $ketsu_max ";
echo "血圧(下): $ketsu_min ";
echo "元気レベル: $level <br>";

mysql_connect('localhost','root','20cuc08!');
mysql_select_db('sotu_a640011');
$sql = "insert into kojindata values('kume', 1, now(),
    $ketsu_max, $ketsu_min, $taion, $level, 0)";
$result = mysql_query($sql);
if (!$result){
    $message  = '無効なクエリです。: ' . mysql_error() . "<br>";
```

```
$message .= 'クエリの内容 : ' . $sql;

echo "<p>";
echo $message;
echo "</p>";
} else {
    echo "正常に登録されました。<br>";
    echo "<a href=taion.html>登録ページに戻る</a>";
}

?>

</body>
</html>
```

図 6.2.3.7 db_touroku.php

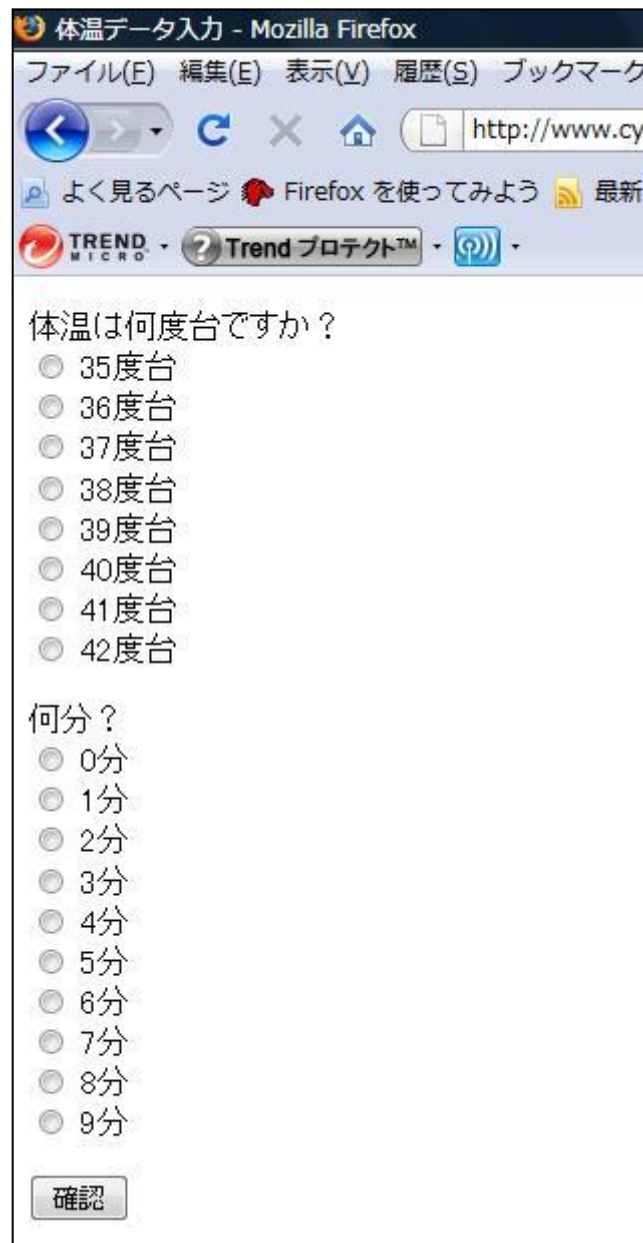
7 章 評価

7.1 機能評価

機能の動作確認

- ・ 初めに、パソコンでデータ登録ができるか試してみる。

taion_touroku.html へアクセスをすると以下のような画面が表示される(図 7.1①-1)



The screenshot shows a Mozilla Firefox browser window titled "体温データ入力 - Mozilla Firefox". The address bar displays "http://www.cya...". The page content includes a menu bar with "ファイル(E)", "編集(E)", "表示(V)", "履歴(S)", and "ブックマーク(B)". Below the menu bar are navigation buttons (back, forward, home, stop) and a search bar. The main content area contains two sections of radio button options. The first section is titled "体温は何度台ですか？" and lists temperature ranges from 35 to 42 degrees. The second section is titled "何分？" and lists time intervals from 0 to 9 minutes. At the bottom of the form is a "確認" (Confirm) button.

体温データ入力 - Mozilla Firefox

ファイル(E) 編集(E) 表示(V) 履歴(S) ブックマーク(B)

http://www.cya...

よく見るページ Firefox を使ってみよう 最新コ...

TREND MICRO Trend プロテクト™

体温は何度台ですか？

- ☐ 35度台
- ☐ 36度台
- ☐ 37度台
- ☐ 38度台
- ☐ 39度台
- ☐ 40度台
- ☐ 41度台
- ☐ 42度台

何分？

- ☐ 0分
- ☐ 1分
- ☐ 2分
- ☐ 3分
- ☐ 4分
- ☐ 5分
- ☐ 6分
- ☐ 7分
- ☐ 8分
- ☐ 9分

確認

図 7.1①-1 体温入力画面

- ・ラジオボタンを選択し、確認を押すと、以下のような画面が表示される。(図 7.1①-2)



図 7.1①-2 体温確認・登録画面

- ・血圧入力へ、を押すと、以下のような画面が表示される。(図 7.1①-3)

体温: 36.5
次に、あなたの血圧の上はいくつですか？数字を押してください。
血圧(上):

1	2	3
4	5	6
7	8	9
0		

図 7.1①-3 血圧入力画面

- ・ 血圧上を数字ボタンで入力後、終了を押すと、以下の画面となる。(図 7.1①-4)

体温: 36.5 血圧(上): 166

次に、あなたの血圧の下はいくつですか？数字を押してください。

血圧(下):

[やり直し](#)

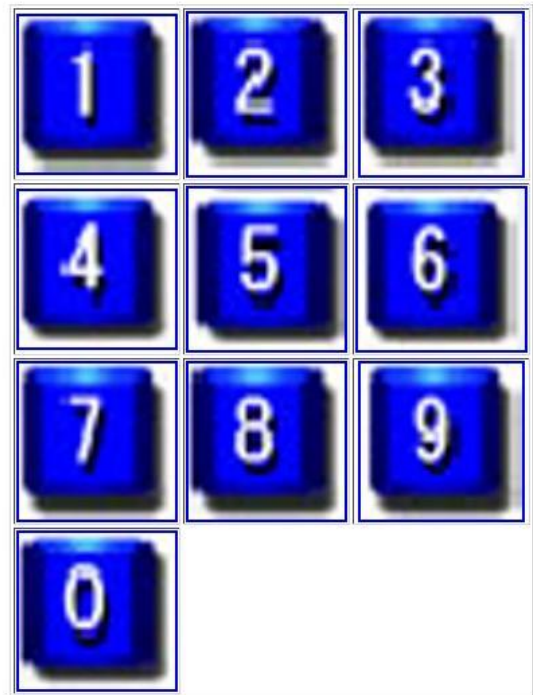


図 7.1①-4 血圧下入力画面

- ・ 終了を押すと、以下の画面が表示される。(図 7.1①-4)

体温: 36.5 血圧(上): 166 血圧(下): 91

次に、あなたの元気レベルはどれですか？

- ☐ 元気
- ☐ まあまあ
- ☐ いつも通り
- ☐ 少し悪い
- ☐ 体調不良

図 7.1①-4 元気レベル入力画面

- ・入力後、終了を押すと、以下の画面が入力される。(図 7.1①-5)

体温: 35.2 血圧(上): 168 血圧(下): 92 元気レベル: 1
正常に登録されました。
[登録ページに戻る](#)

図 7.1①-5 正常登録画面

7.2 考察

当初の目的完了とはいかなかったが、Web を利用し、簡易な入力画面と決められた選択肢の中から選ぶだけという高齢者のために製作したシステムであった。初めてこのシステムを使う人にもわかりやすいシステムとなっている。体温入力画面では、初めから決められた数値を選択するだけで、入力の手間を省き、確認画面で何度何分まで表示し、血圧入力画面では、数字ボタンをゆうちょ銀行 ATM を参考に作成した。最後の元気レベルは、高齢者本人の自覚症状を確認するだけであって、医学的には保障はないが、家族はこれの項目を見て、様子を見に行ったりする事ができるようになってほしい。

今後は、個人情報保護の本人確認システムの開発が必要である。

8 章 まとめと今後の課題

8.1 まとめ

本研究では、デジタルディバイドである高齢者が容易に導入、利用できる形にシステムを目指した。システム製作では、Web 等の汎用的なインタフェースを利用しつつ、画面構成などの工夫により、高齢者が本人一人でも操作可能に限りなく近づいた。

8.2 今後の課題

現在のシステムで登録できる、体温、血圧上下、元気レベルのほかに、心電図などそれ以外の入力項目を増やす。そして、今回の研究で出来なかった本人確認システムを使用できる機能を製作する。また、今回はタッチパネルを使用しなかったが、タッチパネル以外のモニターの使用も検討したい。

9. 参考文献

- ・高室成幸「介護予防ケアマネジメント」中央法規出版 2007 年 p9
- ・川村匡由「情報福祉論」ミネルヴァ書房 2007 年 p29, 41, 65, 148, 149, 150, 151, 152
- ・厚生労働省 平成 19 年度版厚生労働白書 第 2 章 第 2 節 平成 20 年 12 月 28 日
<http://www.hakusyo.mhlw.go.jp/wpdocs/hpax200701/b0040.html>
- ・厚生労働省 平成 15 年度版厚生労働白書 第 2 章 第 2 節 平成 21 年 1 月 15 日
<http://www.hakusyo.mhlw.go.jp/wpdocs/hpax200301/b0005.html>
- ・統計局ホームページ 人口推計月報 年齢, 男女別推計人口 平成 20 年 12 月 6 日
<http://www.stat.go.jp/data/jinsui/tsuki/index.htm>
- ・田中 ナルミ「PHP による Web アプリケーションスーパーサンプル活用編 第 2 版」2008 年 ソフトバンククリエイティブ
- ・西沢直木 「PHP による Web アプリケーションスーパーサンプル 第 2 版」2006 年ソフトバンククリエイティブ
- ・Michele E. Davis 「初めての PHP&MySQL 第 2 版」2008 年 オライリージャパン
- ・Jack D. Herrington 「PHP-プロが教える Web プログラミングテクニック」2006 年オライリージャパン
- ・志村和彦「MySQL データベース構築バイブル」2008 年 毎日コミュニケーションズ
- ・坂井恵「MySQL 辞典」2007 年 翔泳社
- ・山田 祥寛「PHP ライブラリコレクション」 2008 年 翔泳社
- ・日本 ME 学会 BME 1988 年 秀潤社

謝辞

卒業論文の作成にあたり、渡辺 恭人准教授には論文の構成・プログラミング・執筆に至るまで、親身にご指導いただき、感謝いたします。丹羽 宗弘教授には、論文の参考資料・参考文献の提供していただき、感謝いたします。五反田 克也准教授には研究室を使用させていただき、感謝いたします。最後に、私の卒論に関わって下さった方全員にもう一度感謝を述べさせて頂き謝辞とさせていただきます。ありがとうございました。