Dermat – the dermoscopy management system

Grzegorz Surówka¹, Paweł Janik¹, Maciej Ogorzałek¹, Piotr Kanas²

¹Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Kraków, Poland
²The Rydygier Hospital, Kraków, Poland
grzegorz.surowka@uj.edu.pl

Abstract: In order to recognize early symptoms of melanoma, the lethal cancer of the skin, our group is developing pattern recognition and machine learning tools that may help medical doctors in the melanoma diagnosis. Since in the machine learning approach data from diversified sources are required, in this article we present the Java-based Dermat 1.0 application, a medical software system for management of dermoscopy data and the patients follow-up documentation. The chief objective for this system is to integrate all the management activities (image acquisition, anamneses, medical documentation and annotations) into a self-content optimal data base system. Such an integrated approach to digital dermoscopy and management of the patient data between the visits is crucial in comparing the results, storing/retrieving/transmitting dermoscopic images and making a proper and early diagnosis. The Dermat application is distributed among dermatological clinics and private practitioners in accordance to the ‘tool-for-the-data’ model.

Keywords: dermatology, (tele)derm(at)oscopy, patients follow-up, medical software, medical data acquisition and management

1. Introduction

Nowadays computer techniques enter all aspects of life. This is clearly visible especially in medicine, particularly in follow-ups of patients and medical diagnosis [1-3, 28]. Not so long ago hand-made annotations on paper sheets, manual control processes of devices and analog acquisition and management of data were obvious in medical practice. Even now not all medical centers and institutions manage medical documentation fully in a digital way. Especially smaller institutions and private consulting rooms lack good computer-based management systems. This is due to the fact, that the existing commercial software is sometimes dissatisfying and/or too expensive.

Introduction of computer systems into acquisition and management of medical data, apart from its pragmatics, has also more important aspects. Computer driven systems for artificial intelligence and advanced statistical analysis allow for more efficient diagnosis and treatment. Acquisition, storage and retrieving of data in a digital form is a precondition for methods ranging from data mining to telemedicine [1, 29].

The aim of this article is to present requirements for a computer system supporting acquisition of dermoscopy images and follow-up of dermatology patients¹. An instance of such a system is built to help collect dermoscopy and the accompanying data for scientific

¹ This work is supported under grant: N N518 419038 „Zastosowanie technik inteligencji obliczeniowej do analizy danych obrazowych w diagnostyce czerniaka”
investigations within a medical community in Poland. The original motivation to build such a system was to collect good and diverse statistics of the melanoma skin cancer cases to be used in projects aiming at computer aided early diagnosis of melanoma [4, 31].

The piece of software presented here is a database application to store all necessary information about a dermoscopy (or more generally dermatology) patient. It integrates the image acquisition, graphical skin location description, patient follow-up documentation, and management tool for the dermoscopy images into one easy-to-use application.

Some distinctive features of this systems are:

− user-friendly interface,
− reports,
− management of the data base (preparation of selected data sets for scientific purposes, combining data bases),
− complex follow-up system,
− analytical module.

In this work we show first the medical motivation, then the system and functional requirements for the Dermat 1.0 application and its tests, and finally a comparison with the existing tools on the Polish market plus a SWOT analysis for the application.

2. Motivation

Melanoma is a fatal disease of the skin. Due to metastases it is the most malignant human cancer and its high mortality rate is still growing all over the world (references in [4, 31]). Early detection and mass screening is the key of effective treatment of melanoma. One and the most widely used and the cheapest non-invasive method for dermatological diagnosis of the skin lesions is dermoscopy [5]. Dermoscopy (ELM, epiluminescence microscopy) is a medical examination to view the (sub)surface layers of the skin. Instead of visual observation with a bare eye, illumination and magnification of the skin is provided (cameras with additional magnification lenses and enlightenment), thus enhancing classification of morphological structures. Digital dermoscopy (DELM) offers storage of digital skin lesion images and their comparison with the follow-up images (symmetry, color changes). This enhances the diagnostic accuracy of dermoscopy and in the early stages of malignancy is the only method to recognize the suspected skin lesion. Since in DELM computers can collect dermoscopic images, they can also be pattern recognition systems to aid in the course of diagnosing. Diagnostics of the pigmented skin lesions requires much experience to achieve high diagnostic accuracy, so DELM can be additionally used as a telemedicine tool to transmit dermoscopic images to e.g. national diagnostics centers for consulting. For some routine procedures computer aided diagnostics of DELM can be done automatically. The most widely used (video)dermoscopy tools are Dermogenius and MolmaxII [7, 8], some other are also known [9-11].

Dermoscopic follow-up of melanocytic nevi is essential in spotting the changes in the moles over time. This refers to changes in the diameter, asymmetry and colors. An integrated approach to digital dermoscopy and patient data management between the visits is crucial in comparing the results, storing/retrieving/transmitting dermoscopic images and making a proper and early diagnosis.

Our group has worked over machine learning tools for the computer-aided melanoma diagnosis, which may help specialists to recognize early symptoms of melanoma, which are not mature enough to be classified by the well known descriptive methods (ABCD, 7FFM, etc.) [12, 13]. Since this machine learning approach needs learning/testing data from as many as possible diversified sources, we have developed the so called ‘tool-for-the-data’
model, in which the medical community (dermatological clinics and private practitioners) benefits from our cost-free Dermat application to manage their dermoscopy patients, and this tool is used to collect some anonymous data to build training/testing sets for machine learning methods we use. We decided to build such a software system and distribute it into the medical community in Poland to introduce some standards into the acquisition process of dermoscopy data. Unfortunately up to now medical documentation of (particularly) private dermatology practitioners still happens to be stored in ‘ad-hoc’ tools e.g. spreadsheets. This have the following drawbacks:

- data is inserted sequentially as it comes,
- patients follow-ups require separate entries duplicating some data (e.g. personal data, localizations of already diagnosed skin lesions, etc.),
- management of the dermoscopy images is done by separate tools, often by hand – secure association between a spreadsheet entry and its corresponding image(s) requires appropriate labeling of the photographic stuff (e.g. patient’s name, date of birth, date of follow-up, image number),
- some entries belong to strictly defined sets and are standardized (diagnoses, lesion localizations, histopathology scores etc.) so hand-made incorrect annotations in the spreadsheet can produce additional (in reality non-existing) classes/cases of data which makes browsing and statistical analysis of the data difficult,
- preparation (preprocessing) of data for further analyses is difficult and very time-consuming, collection of all data entries and the corresponding photographic documentation for even one patient is a burden,
- reuse of data and external analyses by individuals who were not engaged in the data collection requires time-consuming consultations and assistance.

Thus the objectives for our system are as follows.

- The direct objective is to develop a cost-free, easy-to-use and user-friendly application for the follow-up of the dermoscopic patients.
- The long-range effects and scientific contributions assume:
  - to distribute the application among the dermatological clinics, private practitioners and family doctors in order to integrate the community over the research for the early diagnostics of melanoma,
  - to disseminate in the medical community the standards required in the acquisition process of dermoscopy data,
  - to use the application to collect some anonymous training data for the machine learning methods developed for the sake of computer aided melanoma diagnosis.

Our system is designed and coded under the Object-Oriented methodology, whose assumptions and paradigms are widely known [32, 33] and are not subject to a presentation in this work. Here we get use of those paradigms when presenting the design process of the Dermat application in terms of high-level user interface objects. It reflects the way how medical doctors perceive and express functionalities and their modules/layouts in the application.

3. Requirements for the Dermat system architecture

Dermat is coded with Java to be easily portable to most operating systems. Java is object oriented, robust and secure language (runtime checks of array indices, constant string objects, limited access to protected resources, various security APIs to enhance the build-in security of the language) [24]. JVM (Java Virtual Machine) can be run on different plat-
forms under various operating systems, which makes the application platform-independent. The user interface is coded with use of the Swing packet (Java Foundation Classes, JFC), communication with the database is done through the JDBC interface [25]. JDBC is a low-level interface that makes direct SQL queries: connection to the database, forwarding the SQL instruction to the database, processing of the answer.

Application Dermat (1.0) together with the MySQL [26] database engine and JVM (Java Virtual Machine) requires about 300MB of the disk space. The installation programs have been prepared with the Inno Setup Compiler [27].

There are two database instances in the system. The first one is used for the patients and follow-up data. In this ‘local’ database there has been defined eight tables:

- **patients**: stores personal data of the patients, each patient has a unique entry,
- **appointments**: stores information on the follow-ups, each record correspond to one visit so one patient can collect more than one record,
- **diags**: stores information on anamneses and treatments; one record corresponds to one anamnesis (which is referred to only one lesion location). On one follow-up case several skin spots can be diagnosed thus one record in table ‘appointments’ can be assigned to more than one records in table **diags**. Field id_list_of_changes of this table stays in a relation to table **list_of_changes** (see Fig. 1),
- **photos**: stores data about dermoscopic images. One localization (one lesion) can have more than one entry (record),
- **list_of_changes**: stores the list of diagnosed lesions. Records of this table are in relation 1:1 to records in the table **diags**. This is motivated by the fact that one lesion of a given patient from table **list_of_changes** can be subject to several follow-ups, so medical history of the lesion can be stored,
- **locations**: stores a list of lesion locations. This entry at the user interface allows for picking up fixed names from the standard list,
- **hist_pat**: stores names and codes of the histopathology scores,
- **sources**: refers to the data imported to the actual database. This table stores information on the origin of data and events of data import. The full database structure is presented in Fig. 1.

An extra database (‘management’) in the system manages processes of data import/export. It keeps record of database identifiers, names, servers, ports and user credentials. One can connect to a remote database and use it as a ‘local’ one or merge two or more databases together.

Data export is an important task for the assumed cooperation of medical doctors and our group for the acquisition of anonymous dermoscopic data. The ‘local’ database can be exported to another (MySQL or Oracle) database or a text file and the latter is possible in two ways. Data can either be formatted for humans: in this case columns names are put before data and each data cell is separated with ‘*’, or for applications (i.e. in a binary form) - then only data is stored and each cell is separated by the tab character (‘\n’).

Selection of columns to be sent to the destination table can be done by users. Grouping of data row-wise according to a given parameter and operator (‘=’, ‘!=’, ‘<’, ‘>’) is also possible.

An important requirement of Dermat is language localization. Our application is easy localized through text-based files with menu and medical vocabulary (not hard coded). Now it is adopted to the polish market so in this article it is presented in Polish.
4. Functional requirements

Both the application and the database have been done according to requirements collected during every-day practice of dermatology doctors. Functionally the tool can be divided into two separate modules:

- clinic: it includes patients follow-up, to enter and modify patient data,
- data analysis: graphical presentation of data – to generate and show statistics and correlations.

Advanced utilities e.g. manual database management is described in the next part of this article.

Below we are describing functions and layouts within the modules, which have been thoroughly discussed and designed with the medical doctors. This is a high-level description using GUI objects (italics in the text) reflecting functions and their association states. It should be stressed that the design process, due to the consultations in the medical environment, is the most labour-intensive step of the application development.

Module Clinic

**Entity: New Patient (Dermatology ➔ New Patient or Patients List ➔ New Patient)**

This panel is used to enter personal and medical data for the patient appearing the first time. Each patient can enter the database only once. If there is a need to change their personal data, it is possible through objects Patients List ➔ Modify. Fields of the panel can be subject to changes that are stored in the database in place of the previous values. They are:
Quick Code (a supplementary alias identifying a patient for quick reference; if it is empty, value NULL is entered to the database; this entry must be unique, which is managed by the system); uniqueness is not required so if more than one patient has the same alias, query in a database for that alias brings a group of patient sharing the same alias,

- Name, Family Name, Sex are obligatory fields,
- Date of birth is not obligatory. Since from this value and the current date variable Age is calculated, only numeric values can be entered (other forms are blocked). If not existent or faulty, the NULL value is stored in the database,
- Town, Zip code, Address, Phone number are not obligatory fields. Zip Code is blocked against not-numeric entries.

There is also the ID field on the panel which must be unique in the database (main key) and therefore is not editable.

**Entity: Patients List**

This panel is intended to search for patients and is available through objects Dermatology ➔ Patients List. It has the following fields:

- New: it opens up the New Patient window which is empty,
- Modify: opens up an actualization window which is the same as New Patient window except that is not empty,
- Search: it triggers a query in the database according to criteria defined in this panel,
- Open: opens up a patient record,
- Close: it closes the panel.

The remaining elements of Patients List panel are fields: Quick Code, Name, Family Name, Histopathology Code, Histopathology ID, Localization (this is chosen from a list of standard localization), and Follow-up date. They are used to enter certain criteria according to which patients are searched in the database. Between those attributes the AND condition is used so a query returns patients satisfying all the entered criteria. Leaving all the fields blank results in displaying all the patients.

**Entity Patient Record** consists of three entities: Data, Images, Model.

**Entity: Data**

This is the most important panel containing the medical record of the patient (see Fig. 2). It allows for entering new follow-ups of a case and new medical cases (i.e. new diagnoses). This panel is made up of vertical tabs that represent follow-up visits. Each tab has a date except the last one marked as ‘Today’. The latter has an empty form for the current visit. If it is a new patient, tab ‘Today’ is the only one available.

Each follow-up tab object has three data fields and they are: Data, Images, and Model. 
Data consists of the three forms: Visit, Diagnosis-Therapy, and Lesions.

Visit refers to personal data of the patient and displays Alias, Name, and Family Name (all not editable). The date of the visit can be altered manually, and for the current visit it is taken automatically from the system clock. Field Age is calculated from the date of birth and the current date. It is possible to enter it manually. Consulted by is intended to keep the name of the doctor (clinic) who sent the actual patient to the practice.
Diagnosis consists of the fields: Anamnesis (patient information about the case), Characteristics (description of the skin lesion under interest), Diagnosis (chosen from a list), Lesion dimensions (length, width, depth; the last information can be taken from the hist-pat examination), Localization (not editable), Basis (can be ‘de novo’ and ‘from lesion’), Clark (it applies to pigmented skin lesions where melanoma can develop; it is a list of values I-IV meaning the severity), Metastasis (if cancer is advanced metastases can occur: to another lesion, to organs, or to lymphatic nodes), Breslow (a list of four numeric ranges for the lesion depth; can be available from hist-pat examination), Recrudescence (a list of values: 0/1= no/return of the illness), Survivability (numeric value meaning the number of years after therapy to die; it is known if there is a further information about a given patient).

Therapy has the following fields: Hist-pat ID, Hist-pat code (short abbreviations for diagnosed cases; it is sometimes favorable by practitioners to have a quick reference), Histopathology (a pull-down list to choose the full histopathologic diagnosis), Histopathology contd. (this allows for manual entering of the hist-pat diagnosis if it is not present on the list), Therapy (treatment method e.g. ‘excision’), Surgery (description of the surgery e.g. ‘full’, ‘small lower piece’ etc.), Recipes (to enter the medicines names), and Control (for a note about anticipated follow-ups).

Lesions has a table of diagnosed lesions during actual follow-up. Each lesion is described by its Localization and Diagnosis. There are some buttons that can handle management of lesions: Remove (deletes data about the highlighted lesion from the database), Open existing (this button opens a panel to choose an existing lesion; it allows for additional diagnosis/view of a lesion diagnosed earlier; so it is possible to track development of a lesion during follow-ups), Add new (a button plus pull-down lists to enter a new lesion; localization is chosen by three lists: first we choose a standard localization, then additional description and then some orientation descriptions e.g. ‘left’, ‘up’ etc.; after definitions of the needed parameters and pushing the button the new lesion is visible in the table. This lesion can be then elaborated according to anamnesis and diagnosis). Window Images has a list of images...
present in the table. View of a given image is possible through clicking an entry of the list. The full 1:1 view is possible on the tab *Images*.

**Entity: Images**
This tab serves as a display for skin lesion images in their full resolution. The actual image is highlighted in *Data* → *Images*.

**Entity: Model**
This tab manages the skin lesion images and acts as a precise localization tool to document a lesion site (see Fig. 3 - Fig. 6). To be able to load an image from a digital camera and attribute it to a given lesion ‘automatically’ first spots of the lesions should be marked on the human model. This should happen by clicking the right place with the mouse. The clicked spot is numbered so as to show the sequence of images that must be taken. In that moment on tab *Data* (group *Lesions*) a proper entry is introduced showing the location of the lesion. Of course other crucial data about the lesion should be added by the user afterwards. The right correspondence between images and their descriptions in the application must be supervised by the controller object. Exception objects are cast in case of misalignment of the data entries and the collected images.

The process of loading images by itself is very simple. After making markings on the model skin lesion images are taken from the patients and then the memory card of the camera, or the camera itself is plugged to the PC. Dermat waits for a source of images and loads them automatically to a directory within the application directory. The images get indexed by the system and are attributed to the proper lesion locations. Method for collecting the images presented here is the most flexible tested so it has been implemented.

The human model is shown from all quarters and all parts of the presented body are defined ranges of coordinates. Actual names of the parts pointed by the computer mouse are displayed at the bottom of the window. Switching between parts of body is possible: *Front/Back*, *Sides*, *Head*, and *Palm/Foot*.

![Figure 3. Space orientation: front view and back view](image-url)
A given skin lesion can be associated more than one image (e.g. different shots, side views, magnifications). In the application this is implemented by two lists in the following way. By default the clicked spots are considered to belong to the same skin lesion (usually one lesion is documented with more than one picture) – they appear on the first list while there is an annotation on the other list to start a group. If there is a need to take care of the second lesion on a visit (or more etc.) a New group object is created to collect spots for that lesion. It is possible to come back to editing of the previous image by indicating the proper entry on the first (superior) list.
Each patient is assigned a subdirectory (within the main directory of the collection) with name being the patient ID and their name and family name. In such a subdirectory there are sub-subdirectories assigned to particular follow-ups. Their names are the dates of the follow-ups.

The requirements of the machine learning group for the sake of data acquisition refer to the data import/export. It is useful if there exists more than one instance of the database and the two instances are synchronized with each other. We can easily assume that there is a database in a clinic, and the other similar one in a research center. The main database in a research center can take data from various clinics and private practitioners. Data taken in this way are assigned an individual index to discriminate different sources of data in the further analysis. The export module (Dermatology ➔ Export) activates a configuration panel. To be able to export data to a text file one should specify which attributes should be taken into account (e.g. for personal data protection one can distribute only anonymous data of medical records).

The main elements of the panel are two lists: Available Fields and Exported Fields. The latter one contains those fields that have been chosen from the first list. Data is limited in respect of time by numeric text fields From and To where two time points yyyy-mm-dd are put. Available methods are: Add, Remove, Export, and Cancel.

Import of data (Dermatology ➔ Import) allows for choosing the Available Sources to import from (local file). This is an arbitrary name identifying the data source. If data from a given clinic or a private practitioner are taken for the first time its name should be written to New Source field. In the future versions of Dermat a client-server methodology plus automatic management of import events can potentially be implemented to collect data from clients.
Module Data analysis

The most important innovation in regards to similar tools is the analytical module. It has been assumed that Dermat is equipped with a simple tool (Dermatology → Analysis) for making statistical analyses and visualization of data.

The idea how to visualize data is based on three tabs: Data 1, Data 2, and Source.

Data 1 is used (as the only one) when only one parameter of the data is to be analyzed. The latter parameter is taken from the list of database attributes with additional conditions entered by means of arithmetic operators. In such a graph the abscissa represents values of the chosen parameter from the database and the ordinate shows their statistics (size). It is thus a classical histogram-like chart.

Data 2 panel serves to choose the second parameter of the data for visualization of possible correlations with the previous one. For instance, by choosing ‘localization’ on the first tab, and ‘sex’ on the other tab, we can display a histogram of different localizations regarding men and women separately.

There are common fields of Data 1 and Data 2 classes:
- **Attribute**: a pull-down list to choose an attribute from a database,
- **Value**: a field to define a condition for the latter attribute. For numeric attributes only numeric forms are accepted, for attributes of given values a list of this values is presented.

Individual elements of both classes are:

Data 1:
- **Statistics**: displays min, max, median, and average value of the selected attribute,
- **Data**: opens up a panel with data returned from the database (for the defined query); it allows for exporting data into a text file or different database.
- **Histogram**: displays a histogram chart (X: different classes of the attribute, Y: their size),
- **Point graph**: displays a point-like histogram chart (X: different classes of the attribute, Y: their size),

Data 2:
- **Bubble Graph**: displays a two-dimensional bubble-like chart in that way that on X-axis the attribute from tab Data 1 is presented, and on Y-axis the attribute chosen on tab Data 2. The diameter of the ‘bubble’ depends on the number of pairs (x,y).
- **Histogram**: displays a two-dimensional histogram chart in that way that on X-axis the attribute from tab Data 1 is presented, and the attribute chosen on tab Data 2 is presented by different colors of columns (the number of columns represents the number of classes of second attribute). Axis Y shows here the size of a given attribute class.
- **Point graph**: displays a point-like chart, where on X-axis the attribute from tab Data 1 is presented, and on Y-axis the attribute chosen on tab Data 2. This type of chart is available only if both attributes are numeric.

The third element of the panel Analysis i.e. tab Source serves as an advanced data management system. Usually data coming from different data sources are chosen here. It is possible to define the time period that data had been collected. Data selected/filtered on that tab are displayed as the only ones. The panel has two main elements:
- the names of data sources from which data have been imported (Available Sources),
- the names of data sources taken for visualization (Attached Sources), by default – all data sources are taken.

One can manipulate (add, remove) data satisfying some timely conditions.
Module Data Analysis has also some tools to design individual user’s forms which allow for making any SQL queries on the database. This ‘manual’ way (as opposite to ‘hard-coded’ GUI available by default) makes it possible to adjust the interface and the way one uses the data. This functionality is presented through its use in one of the tests of the application.

5. Application tests

Below we present two classes of tests of the application that emphasize its most competitive aspects.

A) Tests for statistical analyses.

Below we use abbreviations: A-attribute, V-value, and reference to the entities defined in the functional requirements above. Each test is a statistical problem expressed by a user and we show how it is implemented in terms of the application GUI.

Problem 1: The number of women and men in the database.
Implementation 1: Data1: A=Sex → Histogram

Problem 2: Min, max, average and median of women patients.
Implementation 2: Data1: (A=Age, V>0) + (A=Sex, V=w) → Statistics
(w-woman)

Problem 3: Tendency (increasing/decreasing) from year 1997.
Implementation 3: Data1: (A=Year, V>1997) → Histogram

Problem 4: Number of cases versus localization.
Implementation 4: Data1: (A=Localization, V=Head and Neck) + (A=Localization, V=Chest) + (A=Localization, V=Legs) + (A=Localization, V=Palm/Foot) → Histogram

Problem 5: Correlation localization-sex.
Implementation 5:
Data 2: (A=Sex, V=w) → Histogram

Similar correlations: localization-Breslow, localization-Clark, localization-basis can be easily shown.

Problem 6: How many hist-pat types on the Head/Neck.
Implementation 6:
Data 2: (A=Localization, V=Head and Neck) → Histogram
Problem 7: Correlation: X:age-Y:lesion dimensions (taken as a product of width, height and depth in mm).

Implementation 7:

Data 1: (A=Hist-Pat, V=Keratosis Solaris) + (A=Dimensions, V>400) +
Data 2: (A=Age)  \(\rightarrow\) Point Graph

Problem 8: Correlation: X:part of body-Y:age

Implementation 8:

Data 1: (A=Localization) + (A=Age, V>0) +
Data 2: (A=Age)  \(\rightarrow\) Bubble Graph

Similarly other correlations are easy to display: dimensions-the biggest/the smallest, dimensions-localization, dimensions-sex, dimensions-hist-pat, Clark-statistics of classes, Breslow-statistics of classes, metastasis-skin/organisms/lymph nodes, metastasis-hist-pat, metastasis-sex, metastasis-localization, metastasis-Clark, metastasis-Breslow, recrudescence-statistics, recrudescence-hist-pat, recrudescence-Clark, recrudescence-Breslow, recrudescence-localization, survivability-hist-pat, survivability-Clark, survivability-Breslow, survivability-metastasis, survivability-recrudescence.

Some of the cases are shown in Fig. 7-9.

Figure 7. Histogram chart: histological type vs. localization

Figure 8. Point chart: lesion size vs. patient's age
B) Test for data analysis/GUI adaptations.

Tools presented here are intended to design individual user’s forms which allow for making SQL queries on the database and/or adjust user’s interface. To make it clear how to design a user’s form we show an example.

Problem:
Let us design a form to show the content of the database. We assume that in a text field we enter the name of the database, and in the other one the name of the table to be displayed. The action should be triggered with button ‘OK’ which sends the query to the database and then disables the window of the form.

Implementation:
1. We select in the menu: Forms ➔ Creator ➔ New then we choose from the local menu those components which should be located on the form window. The available components are JLabel (to show a message), JTextField (to enter data), JButton (to complete the action).
2. We make all the necessary settings to the components which can be done by clicking on the component with the right mouse button to open the ‘Properties’ dialogue.
3. Action associated with the button can be built with the tab ‘Events’ from dialogue ‘Properties’. We select action ‘Click’ ➔ New and enter its individual name. In the dialogue window ‘Action Creator’ we click ‘Details’ and choose from the list element ‘Connection’ which represents the database connection. From the list of available attributes ‘Variable/function’ we choose send(), and in the text field of the column ‘Value/argument’ we enter:

   !JTextField1 SELECT * FROM !JTextField2
   
where JTextField1 and JTextField2 are the names of the text components that has been located on the form at the label ‘Base’ and ‘Table’ respectively. The exclamation mark in the latter names is intended to distinguish between text and the component names. This action is closed with button ‘OK’. 
4. To create an action that disables the form we select: Action Creator ➔ New. At the component ‘Frame’ we choose method hide().
5. The new form can be saved by Forms ➔ Creator ➔ Save in the system database and is available in menu Forms ➔ Run.
6. Forms designed by a user can be available directly from menu *Forms ➔ Adjust*. With buttons *Add* and *Remove* we choose forms that should be available in the menu.

The above tests show how some of the functionalities of our system are accomplished according to the requirements. They prove not only the developed capabilities and their flexibility but also show clear advantages over the other applications of this type on the market.

In the next section we present a brief comparison with the existing tools and point up all/remaining advantages and disadvantages of our system plus collect its strengths and weaknesses, opportunities and threats in the SWOT analysis.

We deploy our application in the real medical environment by distributing it among private practitioners and dermatology clinics under bilateral agreements. These paths of distributions assume downloading from a project web page (after agreement) or through personal visits in the consulting rooms. There are no formal partners or data centers involved in the process. This reflects the limited customers’ market. The scientific/research interest is yet considerable, which we explained in the motivation.

6. **Comparison with the existing tools**

Medical follow-up and decision support systems are very important to manage the increasing number of data emerging from various clinical procedures and measurements. There is a few applications from the health care domain on the Polish market that support dermatological examinations and patients follow-up documentation [14-23]. They usually have a modular structure and contain a subset of the features mentioned below:

**administration module(s):**
- system configuration/update management,
- user privileges,
- definitions of messages/procedures/services/statistics,
- glossaries of terms,
- International Statistical Classification of Diseases and Related Health Problems databases,
- management of the departments/clinics/branches/partners,
- legal-actions/agreements management,
- health care fund/system insurance accounting/settlements,
- accounting/reports,
- drug lists
- price-lists,
- documents templates,
- calendar of events, personnel time-table,
- patients registration, consultation planning,
- patients/diagnostics flow/queues,
- prescriptions management,
- documentation/data archiving (printers, bar code scanners, chip card readers),

**patient follow up module(s):**
- patient personal data,
- patient history (date/time, anamnesis, recommendations, prescriptions, clinical records),
- examinations and lab tests management systems,
− data import/export (e.g. HTML, Excel SQL, e-mail),
− cross-consultancy,
− special-purpose units (e.g. dentistry, ophthalmology, pediatrics, gynecology, laryngology),

**diagnostic module(s):**
− medical equipment management and data acquisition,
− data pre-view/printout (images/video/recordings),
− management of the clinical data annotations,
− data import/export,

**analytical tools:**
− patients lookup/sorting,
− simple global statistics (sex, age, diseases, locations, prognosis, survival rate),
− simple charts/plots,
− data import/export/merge.

Most of the mentioned Polish commercial systems [14-23] focus on administrative/accounting functions (to satisfy mostly clinics). Chief interests of the commercial applications are business procedures or settlements with the Polish Healthcare Fund, etc. Dermat has no business related modules (‘administration module(s)’) due to its target.

In the current release of our application (1.0) cross-consultancy is not present. All the other branches/interests are covered which is similar or comparable to the commercial tools. The main difference and advantage to the other tools (due to their weaknesses) are:
− the analytical module: complex charts/plots and management of the data base (preparation of selected data sets for scientific purposes, combining data bases),
− the GUI tools to design an individual user interface for the sake of own analytical and graphical requirements,
− charge-free model.

Comparison with the leading dermoscopy management system Photomax [7] shows that Dermat has a more useful (complex) follow-up system. Of course integrated dermoscopic systems (camera+computer+software) have also simplified pattern recognition/computer aided diagnosis tools, so by now they are in advance in this aspect (but it has its price!, Dermat is free of charge). This should be implemented in further releases when enough machine learning data is collected.

We can conclude that there is no one complete and easy-to-use management system which would be suitable for private practitioners, both in functionality (additional statistical/analytical modules) and in price. Dermat outperforms all the commercial systems in this aspect. It covers both administrative and analytical features plus makes it easy to manage the acquisition data to collect and exchange it with scientific centers for further research activities.

7. **SWOT Analysis**

To evaluate the crucial factors of the success of our application we’ve performed a SWOT analysis [30]. We’ve analyzed the favorable and unfavorable as well as internal and external factors: strength, weaknesses, opportunities and threats of the project.

Strengths (advantages over others) are:
– application pricing: free of charge,
– portable size, low physical requirements,
– local, client application,
– an architecture-independent solution (Java),
– no business/special requirements/processes,
– quick time to deployment,
– basic configuration,
– very easy to use (even for no professionals, who do not stay within the medical software constructs),
– compactness: a combination of follow-up management and statistical analysis,
– custom statistics reports (ability to develop own charts),
– access privileges and security roles (sufficient for single location),
– flexible data transfer of images from external sources,
– possible updates/innovations (not obligatory),
– possible interface localization (now: Polish and English),
– quick time-to-value,
– software process
  – use of good practices: factoring, abstraction, commenting, naming, etc.
  – robustness of algorithms and data models,
  – industry standard practices for build, test and release,
  – modularity,
  – design and code efficiency,
  – code security: buffer management, input handling.

Weaknesses (disadvantages relative to others):
– no developed advanced documentation to learn from (instead assistance on-site in case of an agreement between clinic/practitioner and the developers),
– limited performance and scalability tests (huge clinics),
– informal application distribution (from a project web page or through personal contacts), not from multiple, redundant data centers,
– limited software extensibility and customization,
– does not support mobile devices native operating systems,
– problems with market recognition and orientation.

Opportunities (from the environment):
– open source components and possibly open source licensing for the participating teams/coders,
– both team technical knowledge and society/business experience.

Threats (from the environment):
– no developed business partners, no alliance partners, no agency relationships or system integrator relationships,
– no branch certifications,
– no quality certifications,
– customers are not provided a Service Level Agreement (SLA), nor uptime assurance/guarantee,
– no online service or call center,
poor market visibility, strong existing industry leaders controlling the industry messaging and setting the pace for the market,
− some security and privacy risks.

Our SWOT analysis shows that the objective is possible to reach and the competitive advantages are strong enough to market this application in a specialized medical society of dermatology doctors/clinics.

8. Summary and outlook

Nowadays all medical documentation should be stored in a digital way. This is due to time and money savings that come from computer collection and processing. Computer systems can aid the diagnostics process with help of the machine learning and artificial vision. The presented application Dermat 1.0 is a free medical software system for management of dermoscopy data and the patients follow-up documentation aimed at making dermatology practitioners the every-day activities easy. It integrates all the management activities: image acquisition, anamneses, medical documentation and annotations into a self-content optimal data base system. Data that can be collected are consistent and ready to be shared with researchers in accordance to the ‘tool-for-the-data’ model. The Dermat application is distributed among dermatological clinics and private practitioners.

Under the assumption of cooperation of the medical community, the application has sufficient and adequate capabilities to be used as the follow-up and analytical tool for the dermoscopic patients.

There are bright perspectives for new versions of the Dermat application:
− other language localizations,
− client-server mechanisms to send anonymous data directly from the application to the central database through Internet (this could serve also the backup proposes),
− an in-built machine-learning tool for computer aided diagnosis,
− a tool to send lesion images and receive diagnosis in real time from a clinic research centers (cross-consultancy).

References


2 Access date for all the web references is 27.10.2013.
[14] mediquis.gabos.pl
[27] http://www.innosetup.com