

Chapter 11

On the Usage of Health Records for the Teaching of Decision-Making to Students of Medicine

Marcus D. Bloice, Klaus-Martin Simonic, and Andreas Holzinger

Abstract The development of clinical reasoning and decision-making skills in medicine is inextricably linked to experience. Yet students are failing to gain this experience before embarking on their first medical jobs. This is due to several factors, including advances in medical science resulting in patients that are less likely to be hospitalized and more likely to be treated in outpatient departments. This lack of experience within real-world scenarios has resulted in students feeling they are ill prepared for their first medical jobs. One way to counter such a lack of experience is through the use of software simulations such as Virtual Patients. However, simulations are extremely costly to develop, in terms of both financial outlay and the time required to create them. We report here on the development of an iPad-based Virtual Patient simulation system that uses annotated electronic patient data and health records for the creation of cases to enable students to learn critical decision-making skills. By basing these Virtual Patients on abundant patient records, cases can be more quickly and easily created, thus enabling pools of cases to be accumulated—essential for gaining the experience required for the development of sound clinical reasoning skills.

Keywords Decision-making • Virtual Patients • Clinical diagnostic reasoning

M.D. Bloice (✉) • K.-M. Simonic • A. Holzinger
Institute for Medical Informatics, Statistics, and Documentation, Graz General Hospital and
University Clinics, Medical University Graz, Auenbruggerplatz 2, 8036 Graz, Austria
e-mail: marcus.bloice@medunigraz.at; klaus.simonic@medunigraz.at;
andreas.holzinger@medunigraz.at

11.1 Introduction

This chapter describes the development of a Virtual Patient application known as Casebook, which runs on the Apple iPad family of tablet devices. The Casebook application, or app in iOS parlance, teaches students of medicine decision-making through the use of Virtual Patient cases that consist of anonymized and annotated electronic health records. Electronic health records can consist of lab reports, X-rays, electrocardiograms (ECGs), or any other electronically available patient data. This is similar to how Case-Based Learning is performed—where students are guided through a patient case in specially organized seminars and the patient is discussed as they go along. Casebook reads case files which are specially formatted archives of health records that are organized into a timeline of events and examinations that describe a patient’s treatment. Students navigate through these cases and, at certain points throughout the case, must answer questions relating to the patient. These questions normally take the form of “what would you do next?” propositions. When the student has answered a question, the next patient record is revealed thus showing the actual course of events that took place when the patient was treated by the physicians.

This chapter will first describe some of the theories of how to properly undertake the task of teaching clinical reasoning and decision-making. We will introduce what aspects of these teaching methodologies are addressed by the Casebook application and provide arguments as to why the application was designed in this way. Here we discuss why, by using real patient data organized into cases, we can address many of the current teaching methodologies described in recent literature.

Following this, we introduce the Casebook application itself, show how it is used, describe typical usage scenarios, and walk through a typical case presentation. A video of the application in use has also been provided on YouTube to allow readers to experience the look and feel of the application in actual use.

Finally, this chapter will also describe in detail how cases are created, archived, and stored and reports on technical aspects of the development of the application (our group has previously reported on some preliminary prototypes (Bloice et al. 2011a, b)). The ability to easily create, interchange, and archive cases is of paramount importance for the success of the application in terms of the design goals that were stipulated at the beginning of the project.

This chapter concludes with an outlook regarding the future direction of the application’s development.

11.2 Teaching Clinical Reasoning

There is much material regarding the proper teaching of clinical reasoning to students of medicine. A thorough investigation into the multitude of methodologies and strategies used to teach clinical reasoning and decision-making is beyond the

scope of this chapter—however, this section will describe some of the more notable theories that shaped and influenced the design of the Casebook application. By first examining the literature regarding clinical reasoning, and then deciding which aspects of these methodologies could be transferred to a software environment, our group was able to decide how to develop and engineer the Casebook application. This top-down approach meant that our initial requirements were theoretically sound as they were based on well-established and proven arguments from peer-reviewed literature. In other words, this section aims to legitimize and provide the rationale for our design decisions by describing here the literature that guided the application's development.

Judith L. Bowen's review article in the *New England Journal of Medicine* provides an excellent introduction to the strategies that can be employed to promote clinical reasoning in medical education (Bowen 2006). One major theme throughout Bowen's article is the idea that students should be encouraged to develop "illness scripts" which enable for proper diagnostic reasoning. Illness scripts, according to Bowen, can only be developed when previous experience with similar patients enables the student to recognize certain symptoms that point to certain diseases. Without this experience, students find it difficult to translate their theoretical knowledge learned in the classroom to real-world scenarios that students will encounter when they become doctors. Also, experience is invaluable to the clinician as it is the recollection of these experiences that allows them to recognize the subtle differences between similar cases and patient presentations. This argument is reiterated by Norman et al., who state that it is the "power of the plural" which is key to learning diagnostic reasoning and decision-making (Norman et al. 2007).

Traditionally, students have been able to gain experience, and therefore hone their clinical reasoning skills, by dealing with patients while performing ward work during their medical education or clerkships. However, as pointed out by Whitcomb (2006), the changing face of healthcare has resulted in students experiencing less time on wards and gaining less experience with patients with common clinical conditions. This is for several reasons: advances in medical science have resulted in patients being treated in outpatient departments rather than needing to be hospitalized, symptom evaluation is increasingly being performed externally from hospitals (Huang et al. 2007), and students are spending less time on wards due to time and monetary constraints (Young et al. 2009). Simply put, students are seeing less patients as examples of clinical cases (Sanson-Fisher et al. 2005).

Because students are spending less time on wards and are more likely to encounter patients that are exhibiting less common symptoms during their training, it should come as no surprise then that students feel apprehensive about the transition from student to doctor—studies performed by Goldacre et al. (2003) and Cave et al. (2007) have shown that students feel ill prepared for their first medical jobs. According to Goldacre et al. (2003), over 41 % of students either disagreed or strongly disagreed with the statement "My experience at medical school prepared me well for the jobs I have undertaken so far." Respondents to their surveys described that a lack of training that deals with basic problems found on wards and a lack of emphasis on real-life problems were the root of these feelings

of unpreparedness. Similar results were found in Germany, where close to 66 % of doctors felt poorly prepared for work after graduation (Ochsmann et al. 2011).

These aspects are very important in terms of the decision to design Casebook to use real patient records as the basis for creating Virtual Patients. We believe that by basing Virtual Patients on electronic health records, we can tackle these issues from two directions: First, our group believes that by creating pools of similar cases (i.e., cases where each patient exhibits very similar symptoms) students can learn to develop, at least partially, their clinical reasoning skills using this multitude of cases—thus better preparing them for real ward work upon graduation. Second, students feel they are unprepared for their first medical jobs because they have not had enough contact with patients with common conditions. To mitigate this, our group is of the opinion that it must be as easy as possible to generate Virtual Patient cases. If cases can be easily generated or created, then we believe tutors would be more willing to create Virtual Patients that revolve around more common, unexceptional, or simple problems.

Section 11.3 explains in more detail the reasoning behind using patient records to create Virtual Patients.

11.3 Using Electronic Patient Records for Teaching

From the previous section we have seen that learning clinical reasoning can be facilitated through the repeated examination and study of similar patient presentations, in order for students to develop strong illness scripts and to hone diagnostic decision-making. However, what we have also seen is that (a) students feel ill prepared for their first medical jobs and (b) advances in medicine have meant students experience less time on wards and come into contact with fewer patients than ever before. Often, Virtual Patient simulation systems are touted as being good replacements for such situations where it has become more and more difficult for students to experience real-life work and thus improve their perceptions of preparedness and learn proper diagnostic reasoning (Cook and Triola 2009). Indeed, simulations are regarded as “fertile ground” for compensating for exactly this lack of proper experience (Bordage 1999). More recent literature describes ward simulations as particularly suitable in this era of patient safety (Mollo et al. 2012). In fact, much research has shown that simulations do indeed increase retention rates in medical education (Holzinger et al. 2009).

So, although simulations can help to mitigate some of the effects of decreased real-life ward work in medical education, we would argue that most simulations do not in fact do this. A review article published in 2007 by Huang et al. (2007) reported on the usage and costs of Virtual Patients in US and Canadian medical schools. They found that 85 % of individual Virtual Patients cost over \$10,000 to produce, while 35 % of Virtual Patients cost over \$50,000 to produce. They also found that the average time required to produce one Virtual Patient was 16.6 months. It is therefore quite inconceivable that Virtual Patients which are produced or manufactured in

such a way could be built in the numbers required to properly teach students clinical diagnostic reasoning. Again, the proper teaching of diagnostic reasoning and decision-making skills requires that multiple patient cases be available, where each case's patient presentation differs only subtly. Also, as mentioned in the previous section, students complain of not having seen enough patients with common conditions and prototypical patient presentations. Cases which cost \$10,000 are unlikely to be based on common situations or conditions. Nor will the pools of cases that are required to teach clinical reasoning be accumulated when it currently takes over 16 months to create a single Virtual Patient. This has resulted in cases being based on atypical patient presentations, uncommon pathologies, or unexpected reactions to treatment—for the very reason that creating cases for common clinical situations does not warrant the monetary or time investment required. Such cases do not help students who have apprehensions about treating patients with more common conditions and diseases.

Indeed, returning to the work of Bowen (2006), students should be shown patients with prototypical presentations, and then as the student learns to be comfortable with the features of this prototype, they should be introduced to patients with subtle or atypical diagnoses resulting from a similar presentation. Such depth, we believe, cannot be achieved with produced or manufactured Virtual Patients, as they are simply too expensive to create. Therefore, we would argue that creating Virtual Patients—in greater numbers—from electronic health records is a more viable method. Hospital information systems contain a large amount of easily accessible patient records, which, when anonymized, annotated, and organized correctly, could be used to create multitudes of Virtual Patients.

There are other advantages to using real patient data for the creation of Virtual Patients. For example, cases based on real patient data could be used to teach good documentation practices, by presenting students with cases where good practices have been followed. Examples of bad documentation could also be highlighted in a similar way. Likewise, students can be presented with cases where misdiagnosis occurred, to teach them how to avoid diagnosis pitfalls, and so on. Also, colloquialisms and abbreviations that are region or country specific can be taught by showing cases from the university hospital in which they will eventually work. Physicians spend a great deal of time documenting their work. In fact, a survey conducted by Oxentenko et al. (2010) found that nearly 68 % of doctors spend over 4 h per day writing documentation, while only 38.9 % spend this time in direct contact with their own patients. It would make sense, therefore, to maximize the use of patient records and documentation when so much time is invested into this task.

The aim of Casebook, therefore, is to be a platform upon which such Virtual Patient cases can be viewed. Once a patient case has been created by a physician or tutor, the case can be loaded and navigated using the Casebook application. The following sections describe the Casebook application that has been developed and outline in detail how cases can be created from electronic patient records and then viewed using an iPad. Through the use of annotations, physicians can describe their logic and reasoning as well as their thought processes behind making certain decisions. They can highlight values in a lab report which stood out or point to

a symptom or group of symptoms which led to an original working hypothesis. Such annotations are essential insights into the working minds of physicians whose actions students must one day emulate and perform themselves.

11.4 The Casebook Application

The Casebook application itself, which runs on the Apple iPad family of devices, allows for cases, which consist of patient records organized temporally, to be viewed by students who must also answer questions as they progress. In order to facilitate the description of how the application functions, a video demonstration of Casebook in use has been created which highlights some of its features, such as navigating through part of a case and answering questions. The video is available on YouTube under the following URL: <https://www.youtube.com/watch?v=x5R6RZsRTq0>. However, the application's main functionality will also be described in this section, along with a technical description of how cases are imported and displayed, how cases can be stored or archived, and how cases are created.

11.4.1 Using Casebook

Students are intended to use the application in seminar scenarios, either in small groups or individually. Tablet devices, which are omnidirectional and can be passed around easily within groups, make for ideal learning devices for small numbers of students working on a task together. Whether students work alone or in groups does not alter how Casebook is actually used, however. Upon starting Casebook, the student is immediately presented with a welcome screen that displays instructions on how to use Casebook and highlights its main features, as well as describing what each of the buttons on the toolbars do. Because most students will not have seen or used the application before, it was important to offer help from the moment the app starts. These instructions can be scrolled through from left to right and describe the basics of Casebook's functionality. The instructions view is one of four tabs that constitute the application's main front end. Figure 11.1 shows the third tab in the toolbar, which contains the cases that are currently stored on the device.

Upon choosing a case, the student is presented with a number of sections, as seen in Fig. 11.2a. When a case is first opened, only the first section is available to view. Only after finishing the first section may the student begin the second section and so on. When all sections are complete, and the case has finished, the student is presented with the view seen in Fig. 11.2b. Sections could represent hospital departments, visits, stays, and so on.

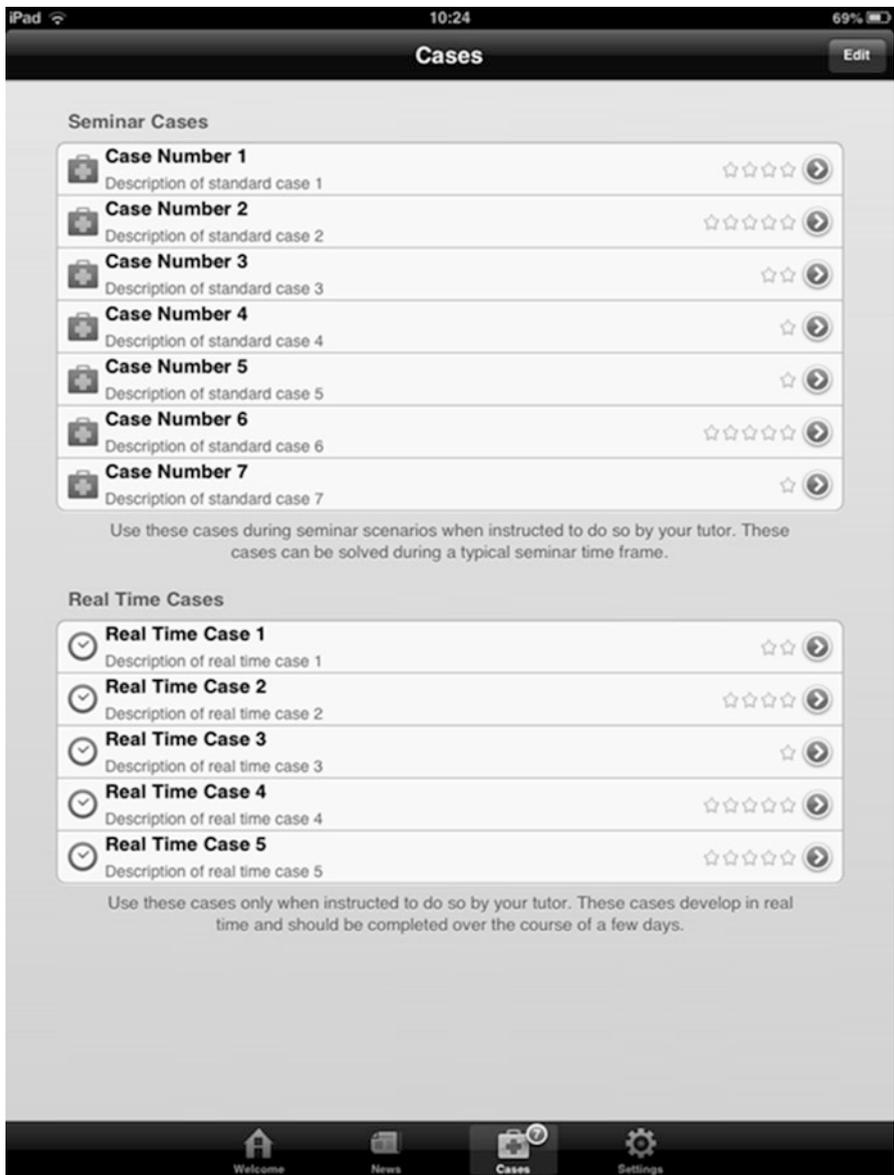


Fig. 11.1 Displaying the cases that have been imported into the Casebook application. Cases are displayed with a rating, given by the student at the end of completing a case. Students should be instructed to rate the case according to how well the case was constructed

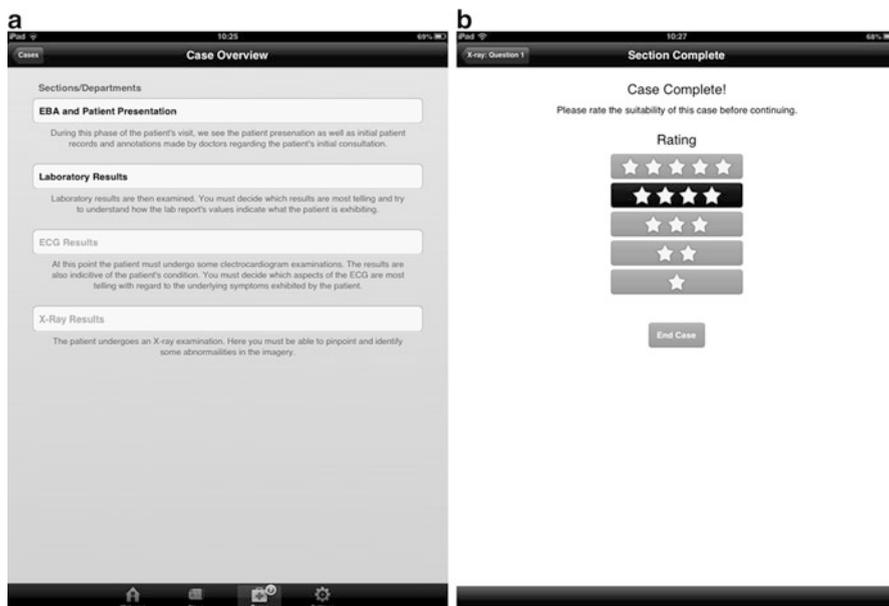


Fig. 11.2 Cases are organized into sections or departments (a). Once a student has viewed and answered the questions of a department, the subsequent group of patient records appears for selection. This follows on in a linear fashion throughout the case until all departments have been completed, at which point the case ends and the student must rate the case before continuing (b)

Once a section has started, a student is generally presented with a patient presentation document or other such health record, as seen in Fig. 11.3a—the order in which patient records appear is stipulated by the case’s descriptor file, described in detail in Sect. 11.4.2. When viewing a patient record, annotations may be available for the student to view. If annotations are available, a small icon on the bottom right of the toolbar is enabled. By tapping on the annotations button, a new screen curls into view showing the annotations for that particular patient record. Figure 11.3b shows such an annotation view for a patient record. The annotation view can be dismissed by tapping on the curled up page on the top of the screen, which returns the user to the original patient record.

Once the student has read the patient record and any annotations that may be available, they should proceed to the next document by tapping on the *Next* button in the navigation toolbar. At various points throughout the case, questions should be answered by the student. Figure 11.4a shows an example multiple-choice question that must be answered by the student before they can continue. Questions can contain hints that can be accessed by tapping on the appropriate icon in the toolbar, as shown in Fig. 11.4b. If no hints are available, the icon is simply disabled. Once the students have selected their answers, they must submit them to check if they

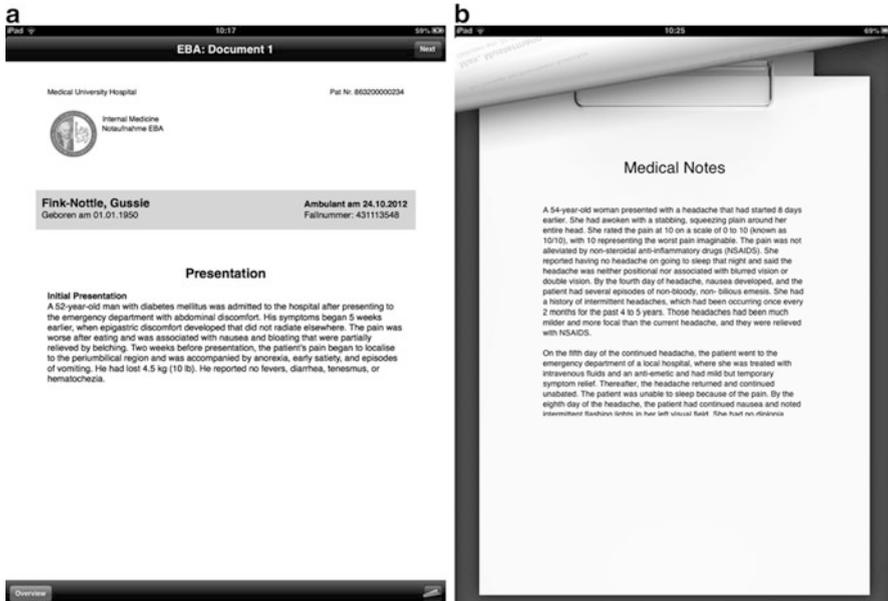


Fig. 11.3 The student views the first patient record in (a). The patient record view contains two toolbars: the toolbar on the top of the view is used to navigate left and right through the patient case. The toolbar on the bottom contains buttons for returning to the list of departments or for accessing annotations. It can be seen in (a) that the annotations icon is enabled on the toolbar, meaning annotations are available to view. Tapping this icon results in the annotations shown in (b). These annotations serve to give students insight into the decision-making and thought processes of the physician

were correct. Students are given immediate feedback, and only after submitting their answers can the student navigate to the next patient record.

This procedure continues until the section is complete. Once a section is complete, the user is returned to the list of sections seen in Fig. 11.2a, where the next department can now be selected. This continues until the student has completed all sections of the case.

Cases themselves are imported into the application by the tutor or professor. The subsequent sections describe how cases are created, stored, and imported into the Casebook application.

11.4.2 Case Format

Cases for use in the Casebook application consist of two main entities:

- A collection of patient records (in PNG format)
- A single descriptor file (meta.json)

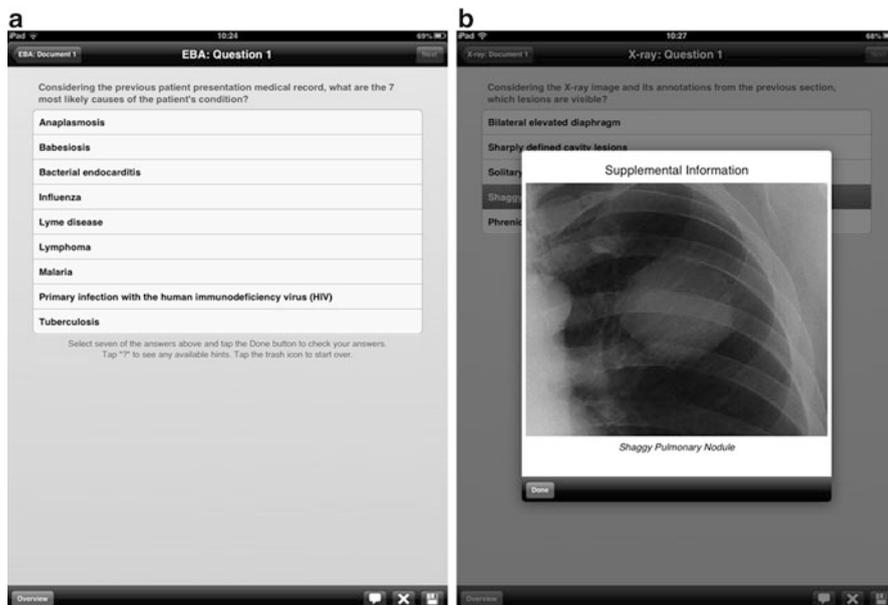


Fig. 11.4 In (a) the student is presented with a multiple-choice question which must be answered before the next patient record can be seen. In (b) hints are shown. If hints are available, the hints button on the toolbar is enabled. Hints may contain images or textual information to help the student when answering questions. Multiple-choice questions can be single or multiple selection

The case descriptor file contains all the case's metadata, including patient record annotations, hints, and question/answer data. The descriptor file, which must be named `meta.json`, is written in the JavaScript Object Notation (JSON) format, an open, text-based standard for information interchange, often compared to XML (Holzinger et al. 2001).

The patient records themselves consist of PNG image files, with each patient record consisting of one PNG image. Multiple-page patient records should be rasterized into one continuous image file, which can be panned and scrolled as if it were a long multiple-page document (see the video demonstration described above). Each patient record is also described using the metadata contained in the `meta.json` descriptor file.

11.4.3 Creating Cases

Currently, cases must be created manually for import into the Casebook application. This is done by first creating a folder containing all the patient records required for the case. There is no requirement for the way in which files are named; however by convention files are named and numbered in the form `<department>100.png`,

<department>200.png, ..., <department>n00.png to make it more clear which files should appear in which order when being viewed in Windows Explorer or the Mac OS X Finder. This convention also allows for files to be placed in between two existing documents if needed, while preserving file order (e.g., eba150.png would then appear between eba100.png and eba200.png in most file managers).

Files should be in PNG format; however, work is underway to allow for PDF files to be viewed directly, using the iOS API's native PDF support. Although there is no minimum resolution requirement, experience has shown that A4-sized patient records that have been exported at approximately 150 pixels per inch allow for more than adequate zooming, panning, and scaling when viewed using Casebook. In terms of performance, PDFs rasterized at over double this resolution displayed, zoomed, and panned effortlessly on a 2nd-generation iPad 2 and first-generation iPad mini—therefore, no performance issues arise even when zooming and panning very large, high-resolution images. Many freely available tools allow for the conversion of PDF to PNG, such as the ImageMagick multi-platform software suite (see <http://www.imagemagick.org/>), or Mac OS X's Preview application. Patient records that contain more than one page (see video demonstration described earlier) should be rasterized into one continuous image file containing all pages of the record.

Once a folder has been created containing the relevant patient records, the meta.json file must be written. This is a metadata descriptor file which contains the case's annotations, hints, and question data.

An extract of such a descriptor file is shown below:

```
{
  "filename": "eba100.png",
  "type": "patientRecord",
  "department": "Outpatients",
  "caseDetails": "Details text that appears before
                  beginning case.",
  "MesHTerms": "D009203",
  "fileOrder": 1,
  "annotations": {
    "annotationHeading": "Sample Annotation",
    "annotationText": "Some annotation text for the
                      file100.png patient record."
  },
  "question": [
    {
      "type": "singleChoice",
      "question": "What is the most likely cause
                  of these symptoms?"
    }
  ],
}
```

```

    {
      "correct": "NO",
      "text": "Answer 1"
    },
    {
      "correct": "YES",
      "text": "Answer 2"
    }
  ]
}

```

As can be seen, the JSON code above describes one patient record, which includes the name of the file (eba100.png) and describes its annotations. The `fileOrder` parameter is an integer value which describes the order in which the patient record should appear. Questions for a particular patient record are defined within that record's metadata.

Once a folder has been created that contains the relevant patient records and a completed `meta.json` descriptor file, the folder is zipped and renamed with the extension `.case`. Once a folder has been bundled as a `.case` file, it is a self-contained case within a file and it is this file that can be imported into the Casebook application using iTunes.

Obviously, it is not ideal to create patient cases manually. Therefore, a Case Creator desktop application has been proposed to facilitate the task of creating cases. A Java-based application would enable a user to drag and drop previously anonymized patient records into the Case Creator interface, organize them as required—into departments and temporally within each department or section—and allow the user to create questions that appear between patient records. The Case Creator would then generate the corresponding `meta.json` descriptor file and output an archived case that can subsequently be imported into the Casebook app using iTunes. It is expected that the application be written in Java for the simple reason that it could be used across several operating systems. However, for the purposes of testing and developing the app according to the requirements collected from the literature research described in the previous sections of this report, priority was the development of Casebook itself.

11.4.4 *Storing Cases*

As described in Sect. 11.2, we believe it is important for teaching institutions to be able to pool cases together, thus allowing for students to view multiple cases of similar patient presentations.

Cases themselves are tagged using MeSH terms. MeSH (Medical Subject Headings) is a controlled vocabulary thesaurus that is used for indexing articles

in the National Library of Medicine's PubMed database. Each case can be tagged with MeSH terms which helps to organize them into categories or to aid searching. This is important for creating pools of cases, where cases that deal with similar themes, diseases, or subjects can be grouped together more easily or automatically. The MeSH vocabulary was chosen as it is freely available, is kept up to date, and has a far smaller footprint than other terminologies such as SNOMED. The 2013 ASCII version of MeSH is 28 MB in size and can be downloaded from the MeSH homepage. When creating cases, educators can access the vocabulary using the online MeSH browser (see <https://www.nlm.nih.gov/mesh/MBrowser.html>) and search for the terms to tag their cases with. The MeSH terms are written into the meta.json descriptor file, along with all other annotations, question data, and so on.

The JSON descriptor format requires both the MeSH Heading and Unique ID be used to tag the cases. For example, a user may wish to tag a case that deals with a patient suffering from the effects of a heart attack with the MeSH Heading *Myocardial Infarction* and the Unique ID *D009203*. For case databases, these MeSH terms could be used to group cases according to MeSH terms—a database of properly tagged cases could easily display any cases that match the ID for myocardial infarction or automatically suggest similar cases depending on previous downloads or search activity (much how PubMed currently behaves when users search for articles).

It would also be possible to create databases of cases that could be accessed via a web-based interface. Cases that are uploaded could be organized according to their MeSH tags and could be searched using MeSH Unique Identifiers or free text by searching each case's MeSH Headings. Being able to properly organize and build pools of cases that concern certain diseases is seen as paramount to the development of the illness scripts described in Sects 11.2 and 11.3.

Cases that have been created can only be transferred to an iPad device using a USB cable and a feature of iTunes known as File Sharing. This is demonstrated in Fig. 11.5. When an iPad device, with Casebook installed on it, is connected to iTunes, the File Sharing tab allows the tutor to drag and drop cases into the application. When the iPad is synced, any selected cases are then transferred to the iPad over USB.

It was a deliberate and intentional design decision to only allow cases to be transferred to the iPad using a USB cable and iTunes, for the simple reason of patient data security. The iOS SDK allows for apps to register themselves as being capable of reading of any particular file format, so that files that are emailed or read over the Internet can be opened by the app. This would make it easier to transfer cases, as they could be emailed to devices. However, email is inherently insecure and many email providers do not support encryption or use SSL or other such security protocols. WiFi networks are also often unencrypted or use older, more easily compromised encryption standards such as WPA (Borisov et al. 2001). Therefore, it is only possible to import cases using iTunes as an interface to the app itself. Files which have been transferred to the iPad device appear in the list of cases, as shown in Fig. 11.1.



Fig. 11.5 Cases can be transferred to an iPad via USB using the iTunes “File Sharing” feature

11.5 Conclusions and Future Work

A number of factors will shape the future work of the Casebook application. There are numerous features which could be included in future versions of the application; it is simply a matter of priority as to which features will be developed first.

Looking into the long term, we plan to create cases that are less linear in structure and more resemble the structure of a flowchart or other such graph-based control-of-flow diagrams. To do so, a description language such as the Oasis Business Process Execution Language (BPEL, see <https://www.oasis-open.org/committees/wsbpel/>) would need to be implemented. Therefore, choices made by students at certain points throughout a case would lead them to different departments depending on what they have chosen. Currently, when a student has completed a section of a case, the next section displayed to them is predetermined and does not change depending on their answers (see Fig. 11.2a). However, it remains to be seen whether cases can be organized in this dynamic way—for example, it makes little sense to allow a student to view a group of patient records that would precede a required examination. Despite this, it would be possible to dynamically rearrange some sections of a case depending on the choices made by the user. The creation of such dynamic and nonlinear cases would require more thought and longer production times than those in a linear structure, yet they would certainly make students feel more like they are in control of the flow of events.

Short-term features include the possibility of using different types of answering methods, such as free text answers. When working with free text, answers can be evaluated in two ways. The first method would be that tutors manually read

all submitted answers and then grade them. The second would involve natural language processing, which has been successfully implemented in the Maryland Virtual Patient (Nirenburg et al. 2009).

Other short-term features will be added to the existing application in order to enhance the user experience. For example, the iOS SDK and APIs are constantly being updated to include new visualization techniques or presentation styles that could be incorporated into the app, such as Collection Views, which make it easier to visualize stacks of objects such as photos, but could be customized to display patient records in groups.

Casebook, in summary, is an application that aims to improve diagnostic reasoning and preparedness by utilizing non-simulated, common, noncomplex, and multiple real-life patient cases within a tablet-based teaching application. The application will be released for free on the Apple App Store upon completion, which will allow other institutions from anywhere in the world to use it within the framework of their current curricular activities. We expect usage scenarios to concentrate on institutions collecting repositories of cases in order to enhance diagnostic reasoning as highlighted in this chapter. However, this is not the only usage scenario for the Casebook system—complex, singular cases could also be used within specialized case-based or problem-based learning seminars.

Patient Data Text data, such as the texts and question data seen in the screenshots, were gathered from The New England Journal of Medicine's Interactive Cases. See <http://www.nejm.org/multimedia/interactive-medical-case>. Radiology images from the Gamuts in Radiology. See <http://gamuts.acr.org/>.

References

- Bloice, M., Simonic, K. M., Kreuzthaler, M., & Holzinger, A. (2011a). Development of an interactive application for learning medical procedures and clinical decision making. In *Information quality in e-health* (pp. 211–224). Berlin/Heidelberg: Springer.
- Bloice, M., Simonic, K. M., & Holzinger, A. (2011b). Using patient records to teach medical students. In *Conference of the Association for Medical Education in Europe (AMEE)* (pp. 230–231). Vienna.
- Bordage, G. (1999). Why did I miss the diagnosis? Some cognitive explanations and educational implications. *Academic Medicine*, 74(10), S138–S143.
- Borisov, N., Goldberg, I., & Wagner, D. (2001). Intercepting mobile communications: The insecurity of 802.11. In *Proceedings of the 7th annual international conference on mobile computing and networking* (pp. 180–189).
- Bowen, J. L. (2006). Educational strategies to promote clinical diagnostic reasoning. *New England Journal of Medicine*, 355(21), 2217–2225. PMID: 17124019.
- Cave, J., Goldacre, M., Lambert, T., Woolf, K., Jones, A., & Dacre, J. (2007). Newly qualified doctors' views about whether their medical school had trained them well: Questionnaire surveys. *BMC Medical Education*, 7, 38.
- Cook, D. A., & Triola, M. M. (2009). Virtual patients: A critical literature review and proposed next steps. *Medical Education*, 43(4), 303–311.

- Goldacre, M. J., Lambert, T., Evans, J., & Turner, G. (2003). Preregistration house officers' views on whether their experience at medical school prepared them well for their jobs: National questionnaire survey. *British Medical Journal (Clinical Research Edition)*, 326(7397), 1011–1012.
- Holzinger, A., Kleinberger, T., & Müller, P. (2001). Multimedia learning systems based on IEEE learning object metadata (LOM). In *World conference on educational multimedia, hypermedia and telecommunications* (pp. 772–777). Tampere.
- Holzinger, A., Kickmeier-Rust, M. D., Wassertheurer, S., & Hessinger, M. (2009). Learning performance with interactive simulations in medical education: Lessons learned from results of learning complex physiological models with the HAEMODynamics Simulator. *Computers & Education*, 52(2), 292–301.
- Huang, G., Reynolds, R., & Candler, C. (2007). Virtual patient simulation at US and Canadian medical schools. *Academic Medicine*, 82(5), 446–451.
- Mollo, E. A., Reinke, C. E., Nelson, C., Holena, D. N., Kann, B., Williams, N., Bleier, J., & Kelz, R. R. (2012). The simulated ward: Ideal for training clinical clerks in an era of patient safety. *Journal of Surgical Research*, 177(1), e1–e6.
- Nirenburg, S., McShane, M., Beale, S., Jarrell, B., & Fantry, G. (2009). Integrating cognitive simulation into the Maryland virtual patient. In *Medicine meets virtual reality* (pp. 224–229). Washington, DC: IOS Press.
- Norman, G., Dore, K., Krebs, J., & Neville, A. J. (2007). The power of the plural: Effect of conceptual analogies on successful transfer. *Academic Medicine*, 82(10), S16–S18.
- Ochsmann, E. B., Zier, U., Drexler, H., & Schmid, K. (2011). Well prepared for work? Junior doctors' self-assessment after medical education. *BMC Medical Education*, 11, 99.
- Oxentenko, A. S., West, C. P., Popkave, C., Weinberger, S. E., & Kolars, J. C. (2010). Time spent on clinical documentation: A survey of internal medicine residents and program directors. *Archives of Internal Medicine*, 170(4), 377–380.
- Sanson-Fisher, R. W., Rolfe, I. E., & Williams, N. (2005). Competency based teaching: The need for a new approach to teaching clinical skills in the undergraduate medical education course. *Medical Teacher*, 27(1), 29–36.
- Whitcomb, M. E. (2006). Ambulatory-based clinical education: Flexner revisited. *Academic Medicine*, 81(2), 105–106.
- Young, L., Orlandi, A., Galichet, B., & Heussler, H. (2009). Effective teaching and learning on the wards: Easier said than done? *Medical Education*, 43(8), 808–817.

Marcus Bloice received his B.Sc. and M.Sc. in Information Technology from the University of Sunderland, UK. Currently he is working at the Institute for Medical Informatics, Statistics and Documentation and is studying for his Ph.D. under the supervision of Prof. Andreas Holzinger at Graz University of Technology. His teaching duties at the Medical University include seminars in decision support systems, medical coding, and medical image analysis. His current research interests include electronic patient records, medical documentation, and problem-based learning in medical education.

Prof. Dipl.-Ing. Dr. Klaus-Martin Simonic is deputy head of the Institute for Medical Informatics, Statistics and Documentation at the Medical University of Graz and a member of the Executive Board of the Austrian Scientific Society for Telemedicine and eHealth. His fields of expertise include clinical informatics, telemedicine, and information quality in the medical domain.

Andreas Holzinger is head of the Research Unit HCI, Institute for Medical Informatics, Medical University Graz, and associate professor at Graz Univer-

sity of Technology, where he teaches Biomedical Informatics at the Institute of Bioinformatics and Genomics and supervising engineering students at the Institute of Information Systems and Computer Media. He started as apprentice in Information Technology, and after his second chance education and IT industry experience, he began his academic studies at the age of 26; studied Information Technology and Physics, Psychology, and Media at Graz University; and obtained his Ph.D. in Cognitive Science from Graz University in 1997 and his Habilitation in Computer Science from Graz University of Technology in 2003. Andreas was visiting professor in Berlin, Innsbruck, Vienna, London, and Aachen. Since 1999 he has participated in leading positions in 30+ R&D multinational projects, budget 3+ MEUR, 300+ publications, 3,600+ citations, h-index =26, and g-index = 76. He is founder and leader of the Expert Network HCI-KDD. Andreas and his team work consistently on a synergistic combination of methodologies and approaches of two areas that offer ideal conditions toward unraveling challenges in handling big and complex data: Human-Computer Interaction (HCI) and Knowledge Discovery and Data Mining (KDD), with the goal of supporting human intelligence with machine intelligence—to discover new, previously unknown insights into the data.