SMU TEACHING BANK: CASE STUDY OF A MULTIYEAR DEVELOPMENT PROJECT UTILIZING STUDENT RESOURCES

Alan Megargel
School of Information Systems, Singapore Management University, ALANMEGARGEL@SMU.EDU.SG

Terence Fan Ping-Ching
Lee Kong Chian School of Business, Singapore Management University, terencefan@smu.edu.sg

Venky Shankararaman
School of Information Systems, Singapore Management University, venky@smu.edu.sg

SMU TEACHING BANK: CASE STUDY OF A MULTIYEAR DEVELOPMENT PROJECT UTILIZING STUDENT RESOURCES

Alan Megargel
School of Information Systems
Singapore Management University
alanmegargel@smu.edu.sg

Terence Fan Ping-Ching
Lee Kong Chian School of Business
Singapore Management University
terencefan@smu.edu.sg

Venky Shankararaman
School of Information Systems
Singapore Management University
venky@smu.edu.sg

Abstract:

A domain refers to a business sector such as banking, healthcare, insurance, manufacturing etc. For an IS student, it is imperative that the domain knowledge includes a comprehension and understanding of business processes, technology and data related to the chosen domain. For example, when learning the retail banking domain, an IS student must have an understanding of the transactions concerned with retail banking such as fund transfers and loan repayments. The student must also gain a strong foothold in transaction fulfilment processes, the various application services that are used, the data that is transferred, etc. Teaching domain knowledge is very difficult, especially when there is no way to put them into practice. One solution to this is through building simulations of the domain where students can perform transactions relevant to the domain. This paper describes a simulated digital bank, "SMU tBank", its architecture, its usage in teaching students, and how it was built using student resources across undergraduate and postgraduate student projects. The paper provides insights on how to structure and deliver useful multi-year pedagogy and possibly other IT projects through effective utilization of only student resources, so that other schools may learn from our experience and further adopt and improve on this model.

Keywords: student resources, multiyear development projects, postgrad, undergrad, pre-university, project experience, guided research, internship, core team, information systems, teaching, banking, classroom labs

I. INTRODUCTION

Singapore Management University (SMU), School of Information Systems, is possibly the only school in the world that is going to the extent of actually building a fully functioning online digital bank, for teaching and research purposes. SMU has embarked on a multiyear programme entitled "SMU Teaching Bank for Financial Services Education", referred to as "SMU Teaching Bank" (or "SMU tBank"). Starting from a clean sheet, SMU is building a fully functioning cloud-based digital banking simulation from the ground up, using today's architecture best practices. The mission of SMU tBank is to become a world class "teaching bank", generating an on-going supply of undergrad and postgrad student projects whereby classroom learning outcomes can be put into practice, leveraging industry-leading banking software and enterprise platforms.

The development of SMU tBank involves three levels of higher education. Postgrad capstone projects specify the solution architecture for banking channels. Undergrad projects deliver banking channel prototypes, based on the solution architecture specified by postgrad capstone projects. Polytechnic (pre-university) students have been engaged through an internship programme to develop components of SMU tBank, which in turn are used to support their own financial technology related diploma courses. Development work started in 2012 and continues today, utilizing only student resources, and without any external funding. This paper presents a case study on how this multiyear development project has been accomplished, utilizing only student resources, and leveraging various project experience course modules.

The main contribution of this paper is to provide one pathway for Information Systems (IS) professors and departments do develop and implement pedagogy and other useful IT projects through the use of student resources. Additionally, this paper presents useful ideas on how "simulation" of real world scenarios can help students learn domain concepts related to business processes and applications in various domains such as banking, health care, supply chain, etc.

SMU Teaching Bank Overview

SMU has obtained academic licenses for several off-the-shelf banking products and middleware products over the years from leading software vendors, in order to support hands-on labs for both postgrad and undergrad courses. SMU tBank is then assembled leveraging this mix of vendor software into a flexible Service-Oriented Architecture (SOA), and student projects then develop banking applications that invoke reusable services.

The Enterprise Platform layer of the architecture is key to enabling the on-going flexibility of SMU tBank as it evolves. The components of the Enterprise Platform layer include; an Enterprise Service Bus (ESB), a Business Process Management (BPM) engine, a Payment Services Hub (PSH), a Business Rules Management System (BRMS), Master Data Management (MDM) services, and an Operational Data Store (ODS).

As a benefit of developing SMU tBank channel applications (e.g. Internet/Mobile Banking, Branch Teller, Payments Gateway, Trade Finance, Conversational Banking AI-driven Chat-bot, etc.), as well as developing the underlying enterprise platform components, students gain a deep technical understanding of how a bank works. Besides gaining banking domain knowledge, students benefit from implementing industry best practices in Enterprise Architecture (EA). SMU tBank is used in the classroom as follows:

In retail banking related courses, students use SMU tBank to learn banking processes such as; account opening, credit evaluation, loan repayments, fund transfers, foreign exchange, standing instructions, General Interbank Recurring Order (GIRO), mobile payments, Two-Factor-Authentication, ATM network management, real-time customer specific promotion offers. Lab questions assess the students understanding of bank processes as well as financial accounting.

In corporate banking related courses, students use SMU tBank financial instruments related to international trade, such as; Letter of Credit, Bill of Exchange, Bill of Lading, Documentary Collection, Trust Receipt, and Export Factoring. Students manage the end-to-end trade process to understand the flow of documents and payments across the relevant parties, e.g.; Importer, Exporter, Freight Forwarder, Issuing Bank, Advising Bank.

In payments related courses, Students use SMU tBank to understand how interbank payments works through an Automated Clearing House (ACH), from different perspectives, a) corporate and retail customers, b) participating banks, and c) central bank. Lab exercises include; payment initiation from corporate customers for both credit transfer and direct debit (GIRO), and bank liquidity management demonstrating scenarios whereby a participating bank has insufficient funds during net settlement with the central bank.

In solution architecture related courses, students use SMU tBank to learn application integration technologies such as message-oriented middleware and web services within an SOA layered architecture as illustrated in Figure 1 below. Labs exercises include; developing services which can be assembled to fulfil complex business logic, and drill-down visualizations of what is actually happening in the services layer when a fund transfer is executed, for example. For their term project, students use the SMU tBank API to assemble their own financial services solutions such as a marketplace lending platform.

Objectives of this Case Study

The objectives of this case study are to; a) raise awareness of SMU tBank as platform for teaching financial technology curriculum at multiple levels of higher education, b) promote SMU tBank as a potential platform for collaboration with multiple universities on related education and practice based research, and c) share with the information systems education community on how a multiyear development project can be accomplished utilizing only student resources, so that other universities may adopt and improve on this model.

II. RELATED WORK

Typically, student projects involve individuals from the same or different cohort on a one-time basis [Jones and McMaster, 2004]. As students typically have year-long programs, and each graduate after several years, multi-year projects necessitate some turnover among the exact individuals involved in the same project over time. This turnover has important deleterious effects on multi-year projects where the results from a previous batch of students become the foundation from which to build for the next batch of students.

Organization scientists have long written about the harmful effect project member turnover can have, countering the effects of increased cumulative experience. The departure of some project members automatically take away with them some knowledge, tacit or explicit, about the project, such that over time, high turnover rates can lead to an organization ‘forgetting’ about what it once knew. Studying the assembly of the Lockheed L1011 aircraft, [Benkard, 2000] documented how incorporating organizational forgetting provides a far better explanation to the observed labour input requirements than simply modelling the effect of reduced productivity as cumulative experience increases (i.e., the learning effect). [Thompson, 2007] provides an analysis on the assembly of the Liberty warship in the U.S. showed that much of the effect of organizational forgetting would be eliminated if the effect of member turnover is explicitly accounted for. These studies highlight the importance of project member turnover in any project.

Similarly, in the context of information systems development (ISD), where a software service company executes on the request of many different customers a variety of projects that are not completely identical to one another, the effect of learning – whereby increased cumulative experience improves performance – is significantly constrained. In particular, studies in this context show that the learning effect is only detected when the same project member takes on the same role across projects, or when there is significant similarity in the technology or application domain from previous projects [Huckman et. al., 2009; Kang et. al., 2017]. While these studies on the effect of learning tend to focus on productivity improvements, reduced labour input requirements in at least one known process can free up important resources for project members to explore alternative approaches to other known problems, thereby aiding the innovativeness of the project team [Postrel, 2002].

The implications from these studies on multi-year student projects are clear. First, any knowledge that is not properly codified will be lost and requires some re-discovering by new project team members. Project team members should be encouraged to document as many ideas as possible, including thoughts, initial back-of-envelop calculations and tentative plans that eventually did not get implemented. Because it is difficult in real life to ask every project member to carry out such a detailed level of documentation, such loss of knowledge is unavoidable.

Second, and as a consequence to the first, practitioners try to minimize such knowledge loss by creating temporally overlapping project memberships, i.e., a subset of project members would overlap with the departure of other members and the arrival of new members. This is witnessed in the television and movie production industry, where members collaborate over multiple projects over time to deepen their expertise and collaborative capabilities [DeFillippi and Arthur, 2002; Manning and Sydow, 2011]. In multi-year student projects, [Cooper and Heinze, 2007] provide an example where junior project members would over time mature to be mentors to new project members. This engineered, temporally overlapping project membership is how student participation in the SMU tBank project is implemented.

III. CASE STUDY

This section covers the evolution of SMU tBank [Megargel, 2018], starting from conception through to present day, with emphasis on how student resources were utilized. The motivation of developing SMU tBank is covered first, followed by an overview of the project development phases. The ecosystem and types of student resources are then covered, followed by the details on how each type of student resource were utilized to develop components of SMU tBank.

Motivation of Developing SMU Teaching Bank

At SMU School of Information Systems, there are two programmes which focus on financial technology; a) Master of Science in Business – Financial Technology and Analytics Track, and b) Bachelor of Science in Information Systems – Financial Technology Track. Prior to the SMU tBank project, both programmes offered very little hands-on experience with financial technology and applications. This lack of hands-on immersive experience with financial technology inhibited students' ability to learn otherwise abstract concepts around banking processes and technology, and FinTech alternative solutions. Furthermore, faculty did not have access to a sustainable and scalable digital banking platform to support ongoing FinTech related research in the areas of; blockchain, microservices architecture, realtime personalized cross-sell, anti-money laundering, KYC / digital identity management, Open API banking, etc. The aim of the SMU tBank project was to provide a fully functioning cloud-based digital banking platform for teaching and research purposes. The platform is now made available to all faculty who are teaching financial technology related courses, and/or are conducting financial technology related research.

Overview of the Project Development

Starting from scratch, having only a set of academic license agreements to use certain commercial banking software and middleware products, SMU embarked on a multiyear journey to architect and build a fully functioning online bank for teaching purposes. SMU tBank employs an SOA layered architecture as illustrated in Figure 1 below, with Service Consumers positioned at the top layer, Service Providers positioned at the bottom layer, and an ESB Framework positioned as part of the middle layer. The ESB (a collection of services) can expose the functionality of the underlying banking systems, or it can implement banking functionality directly as self-contained, separately deployable, microservices.

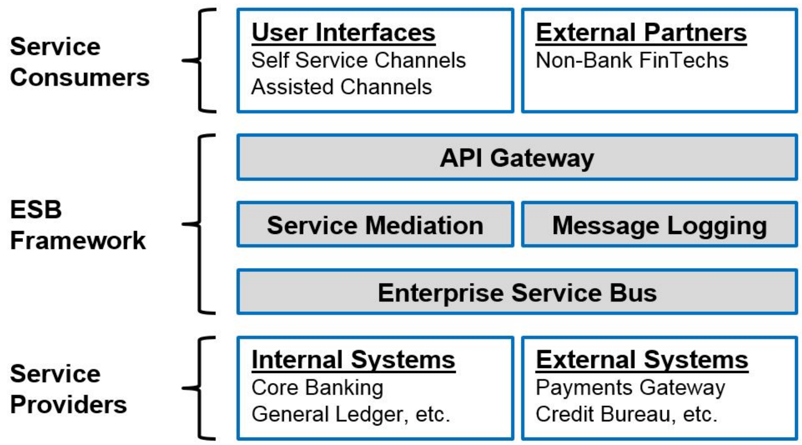

Figure 1: SOA Layered Architecture

Before student teams could be engaged, it was important to establish guiding principles for an SOA Layered Architecture, anchored around an ESB Framework. Once the architecture principles were established, student teams could then be engaged to develop components of SMU tBank starting with core components of the ESB Framework. The implementation timeline for SMU tBank is shown in Figure 2 below.

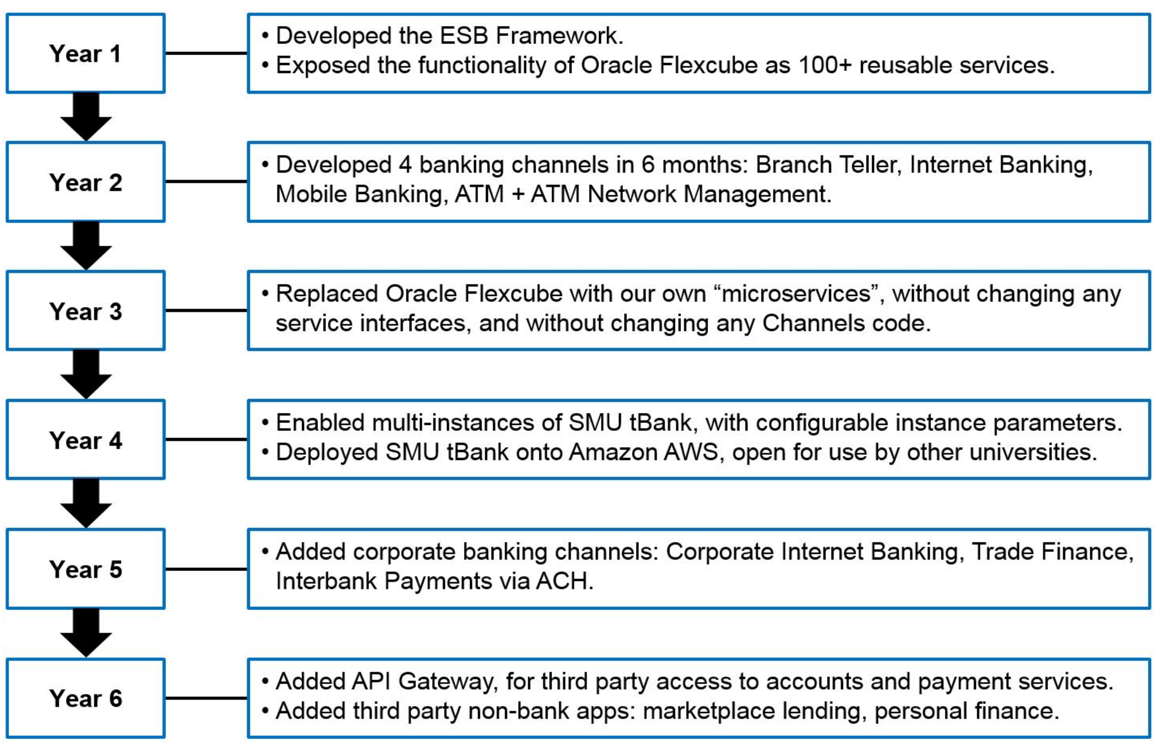

Figure 2: SMU tBank Implementation Timeline

The milestones achieved each year are described as follows:

Year 1 - One student team developed the ESB Framework including the Service Mediation and Message Logging components, the Runtime Governance Tool, and Best Practice Guidelines. Another student team developed over 100 services which exposed the functionality of the Oracle Flexcube core banking system, guided by the Banking Industry Architecture Network (BIAN) Service Landscape, using a GUI-Driven Designer Tool.

Year 2 - Three student teams developed 4 retail banking channels concurrently within 6 months, including; Teller, Internet Banking, Mobile Banking (for both Android and iOS), and an ATM simulation plus ATM Network Management. At this stage, a high level of service reuse had already been achieved, for example the "getAccountBalance" service was reused by all 4 channels, and the actual service consumption could be monitored via the Runtime Governance Tool developed during the previous year. The main learning point for students was that a flexible SOA layered architecture enables agile assembly of new solutions [Megargel et al., 2018], in this case; 4 channels developed concurrently within 6 months. The retail banking capability delivered within 6 months is summarized in Figure 3 below.

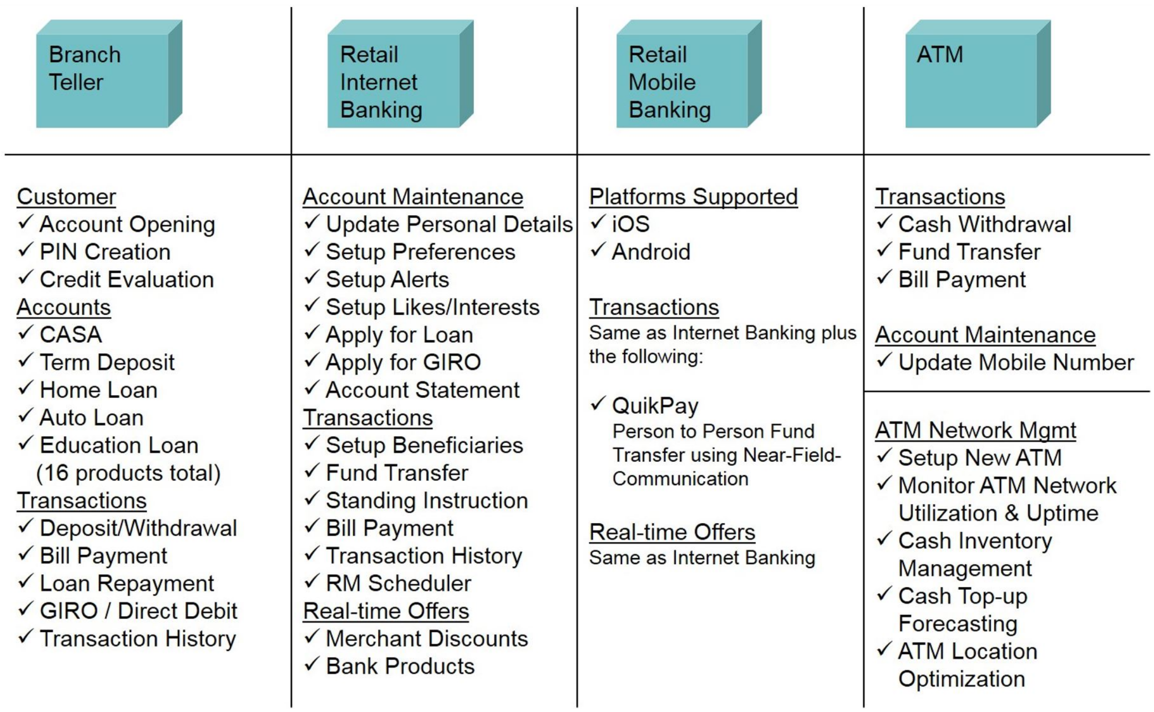

Figure 3: SMU tBank Retail Banking Channels

Year 3 - In preparation for deploying SMU tBank to the cloud, the Oracle Flexcube core banking system (CBS) was replaced with microservices, without changing any service interfaces, and without changing a single line of code in any of the channels applications. This was made possible due to the Service Mediation component of the ESB Framework. Step 1 was to replicate all of the business logic in the CBS into Service Stubs, and to migrate all of the CBS data into those service stubs, as illustrated in Figure 4 below. The service interface, which remained untouched, in combination with the newly developed service stubs, can be considered as self-contained, separately deployable, microservices which own their own data. Step 2 was to use the Runtime Governance Tool to change the channel-to-service mapping in the Service Mediation component, such that the channels invoke the newly created microservices instead of the original CBS-exposed services. Since the service interfaces did not change, the channels were able to continue functioning as normal without requiring any code changes or redeployments. Step 3 was to "unplug" the Oracle Flexcube core banking system. The overall infrastructure footprint of SMU tBank became significantly reduced, which made it feasible to deploy the entire system up onto the cloud. The main learning point for students was that a flexible SOA layered architecture decouples frontend channels from backend core systems, such that large-scale change scenarios such as CBS replacements are simplified.

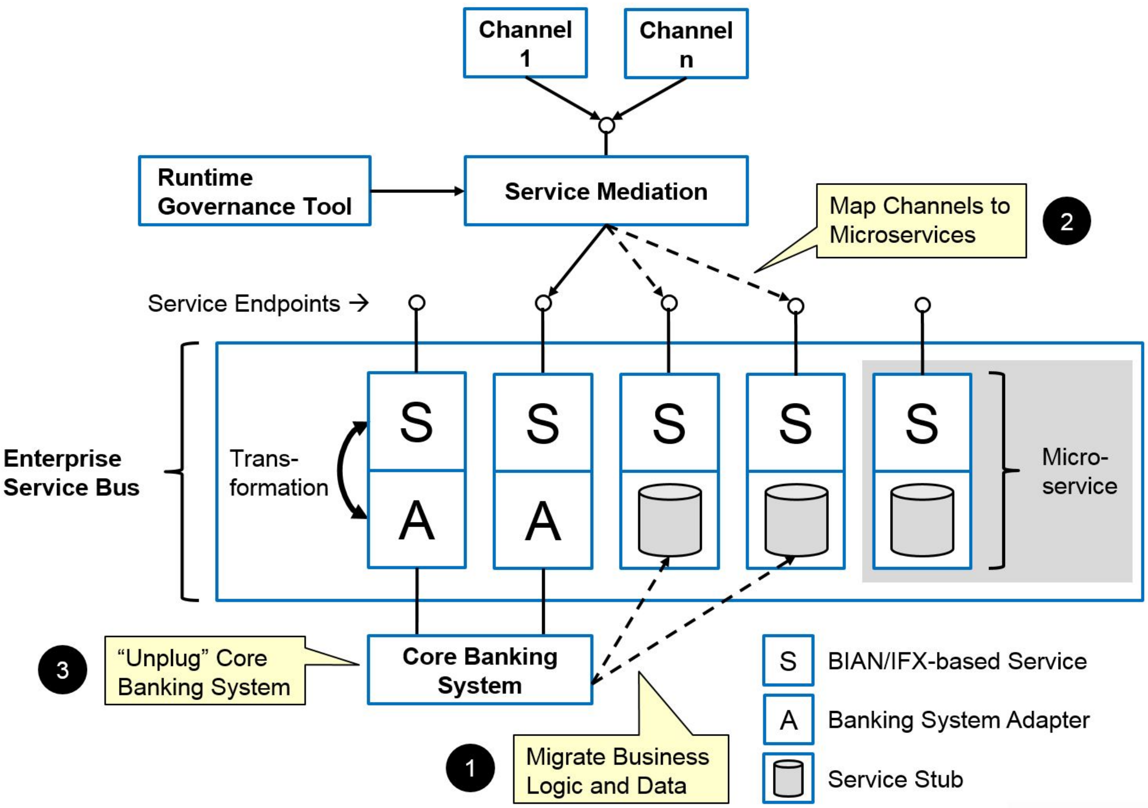

Figure 4: Core Banking System Migration to Microservices

Year 4 - SMU and pre-university polytechnic entered into a licensing agreement for SMU tBank to be used in their Financial Informatics Diploma course. In preparation for this, a student team logically segregated SMU tBank into a multi-instance banking system, with configurable bank-level parameters for each instance, including; base currency, reserve ratio, SWIFT routing number, etc.. Another student team developed a multi-instance General Ledger (GL) system, so that each instance of SMU tBank can have its own GL Chart of Accounts and GL Posting Rules, and can generate separate Financial Reports for each instance of SMU tBank, including; a Balance Sheet and an Income Statement. With the logical segregation of SMU tBank into a multi-instance banking system, the entire SMU tBank suite of software components was then deployed, including the ESB Framework, onto Amazon Web Services (AWS) EC2, and opened it up for use by other universities. During this year, other student teams continued to develop and add components to SMU tBank, including; a Design-time Governance Tool for service lifecycle management, and a Business Rules Management System (BRMS) for centrally managing credit decisioning rules for both retail and corporate products, and realtime inbound marketing rules for delivering targeted and personalised next best offers.

Year 5 - Three student teams concurrently developed; a) Corporate Internet Banking including end-to-end payments processing via a Payment Services Hub, b) Trade Finance including Letter of Credit Issuing Bank and Advising Bank processes and related document and payment flows, and c) an Automated Clearing House (ACH) to facilitate interbank payments between instances of SMU tBank. ACH administration includes; a) registering a bank (SWIFT/BIC code) onto the network, b) setting up a bank's preferred payment message format; SWIFT MT or ISO20022, c) setting up a bank's settlement account with the central bank, d) setting up the settlement schedule with central bank, and e) managing a bank's rules for handling liquidity shortfalls at settlement time, e.g.; prioritize payments by value, prioritize payments by time, overdraft with central bank.

Year 6 - A student team from a pre-university polytechnic, working as interns at SMU, helped to develop and document an API Gateway to provide third party (external) access to SMU tBank accounts and payment services. The same student team then developed a non-bank personal finance app which invokes the SMU tBank API. Another student team from SMU developed a non-bank marketplace lending platform, similar to Lending Club, targeting retail customers. Borrowers apply for loans, and the platform generates a credit score for the borrower based on; provided information, credit bureau report, and past loan performance with the platform. Investors select loans based on risk grade and category, and diversify their risk by investing in small pieces of many loans. The platform invokes services through the bank's API Gateway to execute the loan disbursement and monthly repayments, and the platform takes a fee.

Types of Student Resources

Over the span of 6 years, there were 5 different types of student resources utilized to develop SMU tBank, drawn from 3 different levels of education, e.g.; postgrad, undergrad, and pre-university, as shown in Table 1 below. Except for the SMU tBank Core Team (or "Core Team), all of the student resources contributed for academic credit only, leveraging various project experience course modules, whereby their graded project deliverables were then incorporated into the SMU tBank implementation. None of the student resources were funded by external grants. The Core Team was funded by an internal Work Study Grant. Project durations range from 4 to 6 months each, depending on the course module. Core Team members each committed for a 2 years assignment. A total of 103 students have been involved in developing SMU tBank, including; 13 postgrad, 78, undergrad, and 12 pre-university. One of the challenges of this multiyear project has been to maintain continuity, and implementation knowledge, as project teams roll off and Core Team members graduate and need to be replaced.

Table 1: Types of Student Resources

Level	Resource/Project Type	Description
Postgrad	Capstone Project	Individual Project – 6 months
Undergrad	IS Project Experience	Group Project – 6 months
Undergrad	Guided Research	Individual Project – 4 months
Pre-University	Internship	Group/Individual Project – 5 months
Undergrad	SMU tBank Core Team	Funded via Work Study Grant – 2 years assignment

The different types of student resources/projects, and how each type has contributed to the development of SMU tBank over multiple years are described as follows:

Capstone Project: Postgrad Students Developing Solution Architecture

SMU School of Information Systems (SIS) offers a Master of IT in Business (MITB) – Financial Technology & Analytics Track. The enrolment in this programme consists of a mixture of fresh grads with relevant Information Systems (IS) or Computer Science (CS) degrees, and working professionals from the banking industry. To fulfil their project experience course requirement, students have 2 options; a) internship with a financial institution, or b) capstone project with an external or internal (faculty) sponsor. External internships are paid, while faculty sponsored capstone projects are unpaid. Capstone projects are substantial and therefore are worth 2 course credits (instead of 1). Students who opt to take the capstone project option, may take up an SMU tBank related project, provided that they have relevant industry experience.

The scope of an SMU tBank related capstone project is to deliver a detailed solution architecture (around 100 pages) for one new component/application to be incorporated into SMU tBank. The solution architecture is then handed down to an undergrad IS Project Experience team, which then develops the new application as their project deliverable. Examples of postgrad-delivered solution architectures which get implemented by undergrad project teams include; Internet/Mobile Banking, Trade Finance, General Ledger, Automated Clearing House, Business Rules Engine. One of the challenges of utilizing postgrad students to deliver solution architectures is in matching student interests and aptitudes with SMU tBank needs. Students selected for these projects must already have solution architecture skills as well as banking/FinTech domain experience.

IS Project Experience: Student Teams Developing Applications

The SMU-SIS undergrad degree programme includes a required IS Project Experience course, whereby teams of 5 or 6 students develop and deliver working software applications for external or internal sponsors, for academic credit only. Student teams may select an SMU tBank related project to deliver for an internal faculty sponsor. The development of software applications by undergrad teams is guided by the relevant solution architecture documents previously delivered by postgrad capstone projects.

The undergrad project timeframe is limited to just over one semester (around 6 months), whereby project teams are expected to deliver and deploy fully featured software applications to be incorporated into SMU tBank, e.g.; Internet/Mobile Banking, Trade Finance, etc. Often multiple project teams work concurrently during the same semester. The concurrent development of multiple software applications within the same semester is made possible due to the reuse of existing services within an SOA layered architecture (see Figure 1).

One of the challenges of utilizing undergrad students to deliver software applications is that the level of quality varies from team to team. Once the semester is finished, the project teams roll off. Some features may be left unfinished or may have bugs remaining, which are left to the Core Team to manage. Another challenge encountered, even within this short 6 year’s timeframe, is that javascript frameworks become quickly out of fashion. Recent project teams develop using the latest frameworks, while older applications still in use were developed using what are now legacy frameworks which the Core Team finds challenging to maintain.

Guided Research: Individual Students Developing Advanced Prototypes

The SMU-SIS undergrad degree programme includes an elective Guided Research course which gives qualified individual students an opportunity to support a faculty sponsor with their research, similar to a research engineer role, for academic credit only. Students may select an SMU tBank related research project. In this case, the project deliverable has two parts; 1) an academic style research paper complete with method, results, and conclusion, and 2) an advanced prototype software application which demonstrates the core subject of the paper. Examples of guided research projects completed include; Microservices Architecture Implementation in Banking, and Digital Identity Management Blockchain Use Case for Customer Onboarding. The advanced prototypes delivered by these guided research students are then handed over to SMU tBank related IS Project Experience teams to implement in their software applications.

Internship: Pre-University Student Teams Doing Testing and Documentation

Under the SMU tBank multiyear project, SMU has been running an internship programme with a local pre-university polytechnic. During one semester each year, SMU hires 2-5 students from the polytechnic to work as a team on one SMU tBank project. Polytechnic students earn academic credit for their internship, and are paid a stipend from the SMU tBank project budget. Interns are assigned projects which they can handle, given their limited training at the polytechnic level. Typical internship projects include; documenting the SMU tBank API, developing demo applications which utilize the SMU tBank API, and writing up classroom lab guides. Interns with more coding experience are assigned small application development projects. One of the challenges of utilizing pre-university students to deliver projects is that their technical skill levels vary depending on which diploma programme they are from, and therefore projects must be tailored to match their capability. One benefit of working with this particular pre-university polytechnic, is that they are under a license agreement with SMU to use SMU tBank software to support their own banking related course modules. The internship programme together with the license agreement has strengthened the admissions pipeline between the two institutions.

SMU tBank Core Team: Funded via Work Study Grant

The SMU tBank Core Team is made up of 3 undergrad students in their 3rd and 4th year. They are hired under a Work Study Grant (WSG) where they can work up to 200 hours per semester, and must commit to a 2 years assignment. Every 2 years, a core team graduates, and a new core team is hired to take their place. These students handle all of the ongoing bug fixes and enhancements needed for all of the SMU tBank software applications delivered by IS Project Experience teams since inception of the programme. At present, there are a total of 19 software applications that require ongoing maintenance. Utilizing WSG students to maintain the SMU tBank software applications presents a number of challenges, as follows:

- **Hiring** – New core team members are hired based on two main criteria; a) they must be strong in javascript coding, and b) they must be qualified for WSG, i.e.; they must already be under a student loan or other financial aid programme. It is difficult to identify students with strong javascript skills who are also on financial aid.
- **Legacy** – The core team needs to maintain all of the older SMU tBank software applications which were originally developed using javascript frameworks which are now out of fashion. It is difficult for students to support multiple legacy frameworks.
- **Continuity** – As each core team graduates and is replaced by a new core team, every 2 years, it is difficult to manage the handover from one team to the next, all of the code and implementation details and domain knowledge.
- **Commitment** – While the WSG students are technically employees, their first priority is always their own school work. If some code needs work, it will always take second priority. It is not reasonable to expect a high level of commitment from these students.

Summary of How Student Resources Are Utilized

The SMU tBank multiyear development project relies on an ecosystem of interdependent student participants, as illustrated in Figure 5 below. Undergrad development teams rely on solution architecture delivered by postgrad students. Classroom labs at three different levels of education (postgrad, undergrad, and pre-university) rely on software applications delivered by undergrad student teams. SMU tBank as a whole is reliant on a small core team of undergrad students to handle ongoing software enhancements and bug fixes.

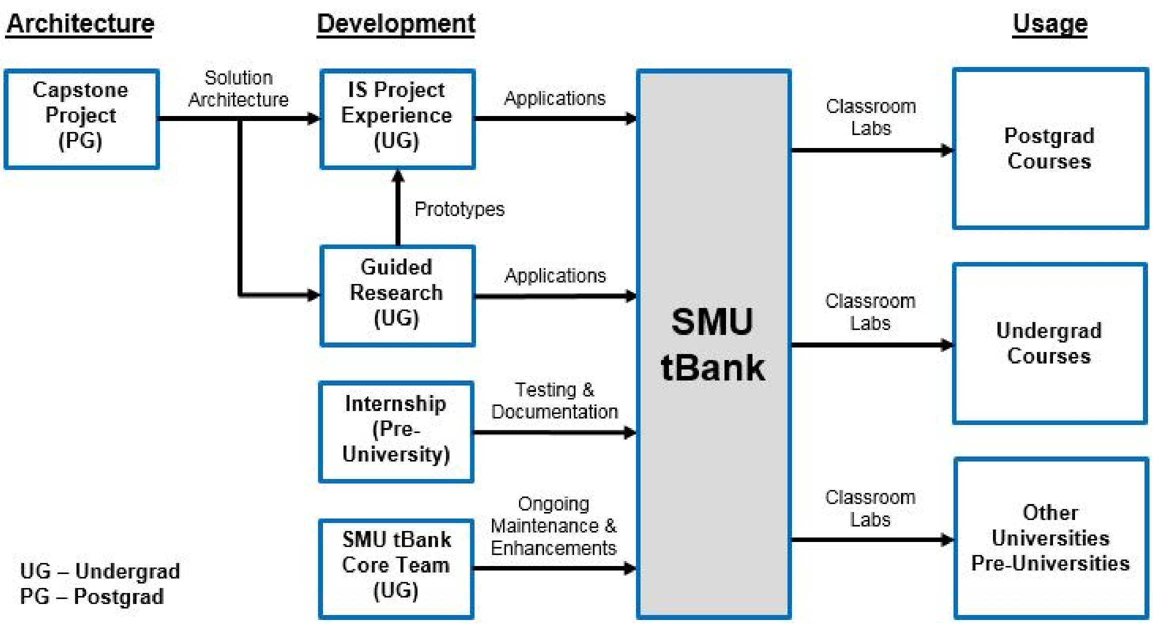

Figure 5: Ecosystem / How Student Resources Are Utilized

Students involved in developing SMU tBank gain a deep technical understanding of how a bank works internally, e.g.; how dual entry accounting works, and how transactions are posted into a general ledger. Students who use SMU tBank in the classroom gain banking domain knowledge of the core business processes, e.g.; customer onboarding, and credit evaluation. Since inception, over 2000 students have benefited from SMU tBank, as summarized in Table 2 below.

Table 2: Number of Students Benefiting from SMU tBank (2012-2018)

Level	**Development of SMU tBank**	**Classroom Usage of SMU tBank**
Postgrad	13	480
Undergrad	78	640
Pre-University	12	800
TOTAL	**103**	**1920**

IV. DISCUSSION

SMU, SIS is possibly the only school in the world that is going to the extent of actually building a fully functioning online digital bank, for teaching and research purposes. The SMU tBank multiyear project is generally successful, given that it has sustained for more than 6 years and has benefited more than 2000 students. The keys to success are summarized as follows:

- Banking industry experience of the faculty sponsor. Specifically, the faculty sponsor should have significant industry experience in an enterprise architecture role.
- Time commitment from the faculty sponsor to manage the project, and support from the Dean to reduce the faculty sponsor's teaching load requirement equitably.
- Availability of students to contribute to the architecture and development of significant software applications, for academic credit only.
- Availability of an internally funded Work Study Grant programme, to cover payment for the SMU tBank core team (of 3 students).
- Cooperation from faculty members to utilize SMU tBank software to support their relevant course modules; e.g. hands-on classroom lab exercises.
- SMU tBank serves as an anchor around which teaching and research outputs are generated, as shown in Figure 6 below.

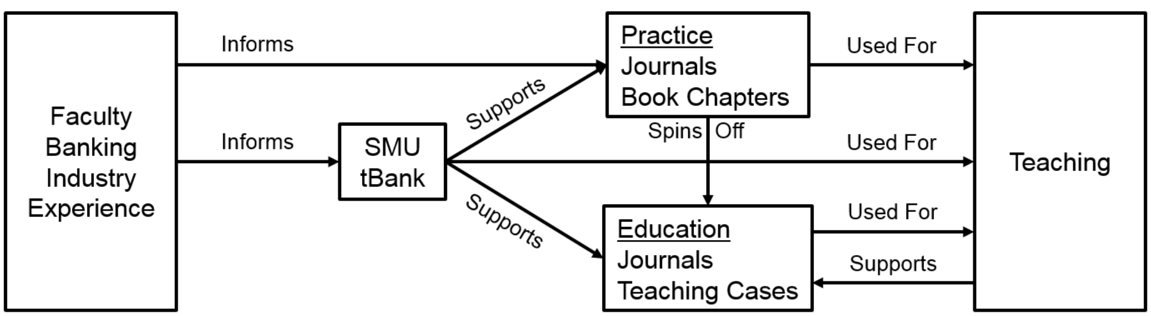

Figure 6: SMU tBank as an Anchor for Teaching and Research

Lessons Learned

While the SMU tBank multiyear project has been generally successful, there remains limitations and challenges that have not been sufficiently resolved. Leveraging an internal work study grant programme to fund student core team members has been problematic for a couple of reasons, namely; a) it is difficult to hire students with the relevant skills who also qualify financially, and b) it is difficult to get commitment from these students because their own school work has priority. A better approach would have been to apply for an external grant to fund fulltime professional developers, perhaps every 3 years, to cover the ongoing software maintenance, and to support the required research outputs associated with the grant.

The problem with maintaining legacy javascript frameworks is perhaps unavoidable. Older software applications, still in use in the classroom, developed using now outdated frameworks still need to be maintained. In hindsight, externally funded professional developers presumably would have been better equipped to maintain legacy frameworks, as compared to students under a work study grant. One benefit, however, of using an SOA layered architecture is that if a legacy application is ever to be replaced, then only the frontend user interface needs to change, because all of the business logic and data exposed via the underlying microservices are fully reusable.

Future Work

Foremost is a goal to invite other universities to collaborate on SMU tBank as a platform for teaching and research, beyond just the one pre-university polytechnic which is currently using it. SMU tBank applications are currently available on the cloud for other universities to use. The SMU tBank API is also exposed via an API gateway, which enables other collaborators to develop their own applications. However, the current level of support from the student core team to support external collaborators is insufficient and is an inhibitor. Therefore, the immediate next step is to seek external funding for a team of professional developers, to support the ongoing maintenance required by both internal and external collaborators.

V. CONCLUSION

An IS student in addition to acquiring technology skills must have a very good understating of the domain where IT solutions are to be implemented and managed. These domain areas can include banking, healthcare, manufacturing, etc. It is imperative that the domain knowledge includes a comprehension and understanding of the business processes, technology and data related to the chosen domain. However, imparting domain knowledge is very difficult, especially when there is no way to put concepts into practice. In this paper we have described a solution that was developed to teach banking domain concepts through the use of a simulated teaching bank, SMU tBank, developed using student resources across undergraduate and postgraduate student projects. The SMU tBank platform has evolved over the last six years from a legacy monolithic architecture to a modern microservices-based architecture. Over a hundred students have been involved in the development effort and around two thousands students have used SMU tBank to learn banking domain concepts and technology. In this paper we have shared our journey through the development and usage of SMU tBank across undergraduate and postgraduate courses. More importantly, this paper provides useful insights on how to structure and deliver useful pedagogy and other multi-year IT projects through effective utilization of only student resources, so that other schools may learn from our experience and further adopt and improve on this model.

ACKNOWLEDGEMENTS

The authors would like to thank the faculty and admin staff of the School of Information Systems, Singapore Management University, who have supported this project over the years. Mostly, we would like to thank the hundreds of students who have directly contributed to the development of SMU tBank, and are now benefiting from this experience as professionals in industry.

REFERENCES

Benkard, C. L. (2000). Learning and forgetting: The dynamics of aircraft production. American Economic Review, 90(4), 1034-1054.

Cooper, G. S., & Heinze, A. (2007). Centralisation of assessment: Meeting the challenges of multi-year team projects in information systems education. Journal of Information Systems Education, 18(3), 345-356.

DeFillippi, R. J., & Arthur, M. B. (2002). 15 Paradox in Project-based Enterprise: The Case of Film Making. In *Managing Innovation and Change* (2nd Ed), Eds J. Henry & D. Mayle. Sage: London, UK: 189-200.

Huckman, R. S., Staats, B. R., & Upton, D. M. (2009). Team familiarity, role experience, and performance: Evidence from Indian software services. Management science, 55(1), 85-100.

Jones, M. C., & McMaster, T. (2004). Addressing Commercial Realism and Academic Issues In Group-Based IS Undergraduate Project Work. Journal of Information Systems Education, 15(4): 375-381.

Kang, K., Hahn, J., & De, P. 2017. Learning effects of domain and technology knowledge in outsourced information systems development: An empirical study. Information Systems Research, 28(4), 797-811.

Manning, S., & Sydow, J. (2011). Projects, paths, and practices: Sustaining and leveraging project-based relationships. Industrial and Corporate Change, 20(5), 1369-1402.

Megargel, A. (2018). Digital Banking: Overcoming Barriers to Entry (Doctoral Dissertation). Retrieved from Singapore Management University: https://ink.library.smu.edu.sg.

Megargel, A., Shankararaman, V., & Fan, T. P. (2018). SOA maturity influence on digital banking transformation. *IDRBT Journal of Banking Technology*, *2*(2), 1.

Postrel, S. (2002). Islands of shared knowledge: Specialization and mutual understanding in problem-solving teams. Organization science, 13(3), 303-320.

Thompson, P. (2007). How much did the Liberty shipbuilders forget?. Management science, 53(6), 908-918.

ABOUT THE AUTHORS

Dr. Alan Megargel is an Assistant Professor of Information Systems (Practice) at Singapore Management University where he serves as Coordinator for the undergraduate Financial Technology Career Track. His current areas of specialization include; enterprise architecture in banking, service-oriented architecture (SOA), payments technology, and non-bank FinTech alternative financial services. Alan has 30 years of industry experience including; Chief Technology Officer at TIBCO Software Asia, Vice President and Head of SOA at OCBC Bank, and Senior Enterprise Architect at ANZ Bank. His banking technology experience covers; Retail & Corporate Banking, Basel II, Data Warehouse, Data Centre Operations and Technology Infrastructure. Alan holds a Doctor of Innovation from Singapore Management University, and a Master of Science in Software, Systems and Information Engineering from the University of Sheffield.

Dr. Terence Fan Ping-Ching is an Assistant Professor of Strategic Management (Education) and Academic Director of Accreditation at the Lee Kong Chian School of Business, Singapore Management University. He holds a PhD in Engineering Systems from Massachusetts Institute of Technology, USA. His current areas of specialization include strategic analyses, especially on the transportation industry. His work has been published in journals including Strategic Management Journal, the Journal of International Business Studies, Management International Review, Transportation Research A – Policy and Practice; and Transportation Research E – Logistics and Transportation Review.

Dr. Venky Shankararaman is a Professor of Information Systems (Education) and Deputy Dean (Practice & Education) at the School of Information Systems, Singapore Management University. He holds a PhD in Engineering from the University of Strathclyde, Glasgow, UK. His current areas of specialization include business process management and analytics, enterprise systems architecture and integration, and education pedagogy. He has over 25 years of experience in the IT industry in various capacities as a researcher, academic faculty member, IT professional and industry consultant. Venky has designed and delivered professional courses for government and industry in areas such as business process management and analytics, enterprise architecture, technical architecture, and enterprise integration. He has published over 75 papers in academic journals and conferences.